NASA Technical Paper 1513

Flight Tests of the Total Automatic Flight Control System (TAFCOS) Concept on a DHC-6 Twin Otter Aircraft

William R. Wehrend, Jr., and George Meyer

FEBRUARY 1980

NASA

NASA Technical Paper 1513

Flight Tests of the Total Automatic Flight Control System (TAFCOS) Concept on a DHC-6 Twin Otter Aircraft

William R. Wehrend, Jr., and George Meyer
Ames Research Center
Moffett Field, California

National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1980

TABLE OF CONTENTS

SYMBOLS

A_{av}	direction cosine matrix between body and velocity axes
A_{cs}	direction cosine matrix between commanded and inertial axes
A_{vsc}, A^{t}_{vsc}	direction cosine matrix between velocity and commanded axis system and the transpose of this matrix
C_{DC}	commanded drag coefficient, $\frac{\text{drag force}}{\text{EQS}}$
C_{LC}	commanded lift coefficient, $\frac{\text{lift force}}{\text{EQS}}$
$C_{mc}(i)$	commanded total moment coefficient, three components, $\frac{\text{moment}}{\text{EQSC}}$
C_{TC}	commanded thrust coefficient, $\frac{\text{thrust}}{\text{EQS}}$
$C_{VC}(i)$	commanded total force coefficient, three components, $\frac{\text{total force}}{\text{EQS}}$
c	wing chord, m
d( )/ds	path length derivative
EA_{as}	direction cosine matrix of estimated body attitude with respect to inertial space
EFV	estimated total acceleration for vehicle as measured by accelerometers, m/sec^2
$E_i(-)$	direction cosine matrix for single rotation of angle (-) about axis i
EMASS	estimated aircraft mass, kg
EQS	estimated total dynamic pressure times wing area of aircraft, N
$ER_s(i)$	estimated vehicle position from the navigation routine, three components in inertial reference frame, m
$EV_{rwsc}(i)$	commanded velocity with respect to the wind, three components, m/sec
$EV_s(i)$	estimated inertial aircraft velocity, three components, m/sec
$E\alpha$	estimated angle of attack, deg
$E\beta$	estimated angle of side slip, deg

$E\delta_A$, $E\delta_E$, $E\delta_R$	measured aileron, elevator, and rudder positions, deg
F_{VCI}	commanded acceleration from command generator, m/sec^2
F_{VI}	commanded acceleration input to trimmap, m/sec^2
g	acceleration due to gravity, m/sec^2
K_{ATC}	gain used in integral operation for commanded acceleration
K_{reset}	reset gain
L	delay of one cycle
NWP	next way point
R	turn radius for commanded trajectory circular arcs (ATCGEN), m
R_{SC}, V_{SC}, $\dot{V}_{SC}$	commanded position, velocity, and acceleration from trajectory command generator measured in inertial coordinate system, m, m/sec, and m/sec^2, respectively
R_{SS}, V_{SS}, $\dot{V}_{SS}$	commanded position, velocity, and acceleration from trajectory command generator measured in inertial coordinate system, m, m/sec, and m/sec^2, respectively
S_{ATC}	distance along path commanded by ATCGEN, m
S(-)	skew symmetric matrix for quantity in brackets
T_s	throttle handle command, m
$\dot{T}_s$	throttle rate command, m/sec
u_s	unit vector defining direction of ATCGEN commanded path
V_{ATC}, $\dot{V}_{ATC}$	velocity and acceleration commanded by ATCGEN, m/sec and m/sec^2
V_{CAL}	calibrated airspeed measured by air data equipment on aircraft, m/sec
V_c	magnitude of velocity commanded by trajectory command generator, m/sec
WPN	next way point
α_c, β_c, ϕ_{vc}	commanded angle of attack, side slip, and roll about the velocity vector commanded by the attitude command generator, deg
α_s, β_s, ϕ_{vs}	commanded angle of attack, side slip, and roll about the velocity vector, commanded by trajectory loop, deg

γ_c, ψ_c	velocity vector glide slope and heading commanded by ATCGEN, deg
γ_v, ψ_v	velocity vector glide slope and heading commanded by trajectory command generator, with respect to the airmass, deg
Δ_A, Δ_E, Δ_C	commanded change in aileron, elevator, and rudder commanded to aircraft by TAFCOS, deg
δ_A, δ_E, δ_R	aileron, elevator, rudder position, deg
δ_F	flap setting measured on aircraft, deg
δ_i	column matrix with 1 in the ith row and 0 in the others
$\delta( )/\delta( )$	partial derivative
ϕ, θ, ψ	Euler angle defining aircraft attitude, deg
ω_c, $\dot{\omega}_c$	angular rate and acceleration commanded by attitude command generator, rad/sec and rad/sec^2
ω_s, $\dot{\omega}_s$	angular rate and acceleration commanded by trajectory loop, rad/sec and rad/sec^2

FLIGHT TESTS OF THE TOTAL AUTOMATIC FLIGHT CONTROL SYSTEM (TAFCOS) CONCEPT ON A DHC-6 TWIN OTTER AIRCRAFT

William R. Wehrend, Jr. and George Meyer

Ames Research Center

SUMMARY

A program has been conducted at Ames Research Center over the past few years to develop a new flight control concept, one that would provide an integrated control for vehicles with difficult nonlinear and highly complicated control problems. The result is a control concept called TAFCOS, for total automatic flight control system. The fundamental idea in the design of TAFCOS is to make maximum use of a priori knowledge of the vehicle characteristics and to build that information into a structure that permits control of the vehicle over the entire flight envelope, without the need for complex mode-switching logic. This report describes the first flight test of the concept, in which flights were conducted using a DHC-6 Twin Otter, an aircraft that is equipped with a digital flight control system. To evaluate the TAFCOS concept, the aircraft was controlled while flying a variety of speed and altitude changes, including a 6° STOL approach, thus exercising a good part of the aircraft flight envelope. The main objective of the flight test was to verify that the TAFCOS structure is suitable for use in a typical digital flight system and that the computational structure has the ability to cope with a real-world environment. The implementation of TAFCOS in the digital flight system showed that the integrated nature of the TAFCOS concept provided a compactness that gave a considerable advantage in usage of computer space and time over a conventional controller design with equally good performance. The flight-test results also demonstrated the capability of the concept in the face of sensor noise, air turbulence, and uncertainties in the knowledge of the aircraft model.

INTRODUCTION

With the development of the new aircraft, such as STOL and VTOL vehicles and those using fly-by-wire and active control concepts, a need was seen for the development of an integrated flight control system that would have the capability of providing the level of control that those aircraft required. A flight control concept called TAFCOS (total automatic flight control system) was developed in response to that need. A theoretical description of the concept is given in reference 1. The control problem presented by the STOL and VTOL vehicles is that they generally exhibit coupled nonlinear flight characteristics; in addition, they usually have a considerable degree of control redundancy. Because the conventional flight controller design requires the use of mode-select options and gain scheduling to handle the control problem, there is a likelihood of achieving good performance only on the design points

with degraded performance elsewhere in the flight envelope. The active control and fly-by-wire vehicles require the flight controller to operate over the entire flight regime of the aircraft; a conventional design may not have the capability to do so. The source of the difficulty with the conventional controllers when applied to active control and fly-by-wire aircraft seems to be twofold: the controller structure does not permit the inclusion of sufficient detail about the vehicle to be controlled and its logic structure is not powerful enough to properly use such data. The structure of TAFCOS was conceived as an integrated flight controller to resolve just such difficulties where the design makes use of detailed a priori knowledge of the vehicle characteristics. That information is built into a logic structure that will permit control of the vehicle over the entire flight envelope. The result is a system that will handle nonlinear flight characteristics without the need for complex mode-switching logic. This report describes the mechanization of TAFCOS for simulation and flight testing and presents the flight-test data.

The flight test was conducted with a DHC-6 Twin Otter aircraft, an aircraft that is equipped with a digital avionics system called STOLAND (see refs. 2, 3). The mathematical structure of TAFCOS is such that a digital system is necessary to carry out the required control operations. The STOLAND system provides the necessary computational capability and a complete hardware/software flight package, which includes support software routines, such as navigation computations, display operations, and other computer housekeeping functions. To carry out the flight-test program, the TAFCOS control laws were first programmed on an IBM 360 for an evaluation of the structure of the controller operation as specifically set up for the Twin Otter aircraft. The IBM 360 used a FORTRAN IV program language and operated with the same model of the aircraft as was later used in the evaluation with the STOLAND system. The TAFCOS program was then converted to assembly language form for use in the STOLAND system, and reevaluated in a flight simulator. This flight simulator uses the flight hardware components as a part of the simulation setup so that this particular task debugged and verified the program that was later to fly in the aircraft. The TAFCOS program was merged with the required portions of the STOLAND software, and various displays and pilot interfaces were brought into operation. The TAFCOS system was then flown on the Twin Otter.

The information presented on the flight test is divided into several sections. The main body of the report consists of a description of the theory underlying TAFCOS, a description of the simulation and flight-test equipment, and a presentation of a summary of the flight data. More detailed information is presented in the appendixes where a complete description of the mathematical structure of TAFCOS is given as specifically applied to the Twin Otter flight test (appendix A); the software structure is described (appendix B); and a complete set of flight data is presented (appendix C).

TAFCOS STRUCTURE

The basic rationale behind the structure of TAFCOS can be seen by considering figures 1 and 2. Figure 1 gives a conceptual outline of a conventional-type controller. The vehicle characteristics are given by the block marked

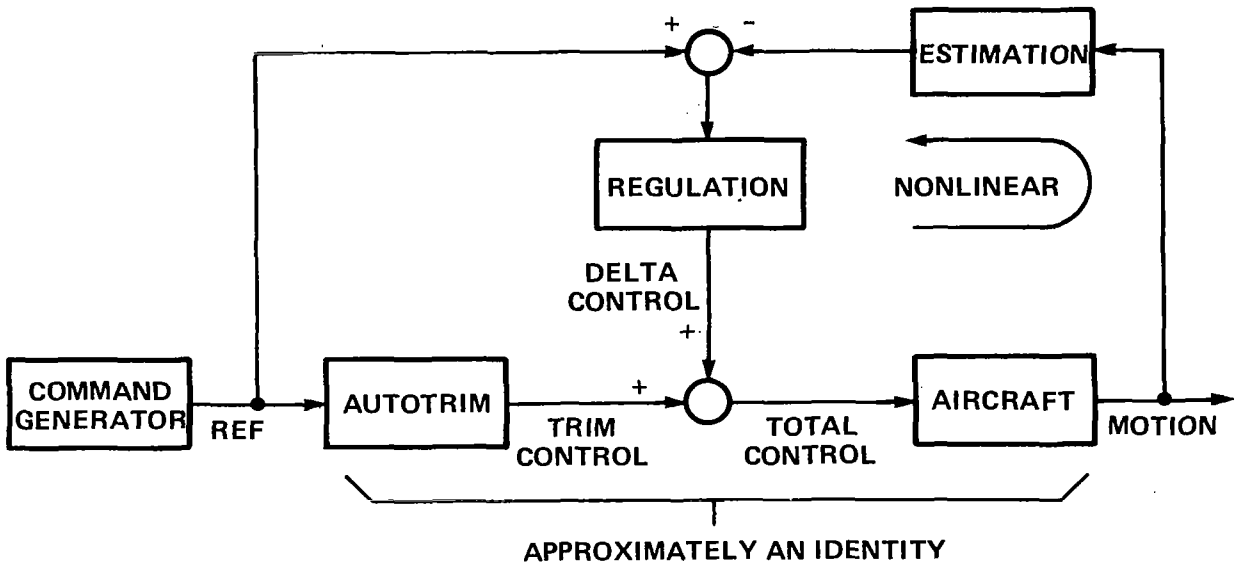


Figure 1.- Usual arrangement of a conventional-type controller.

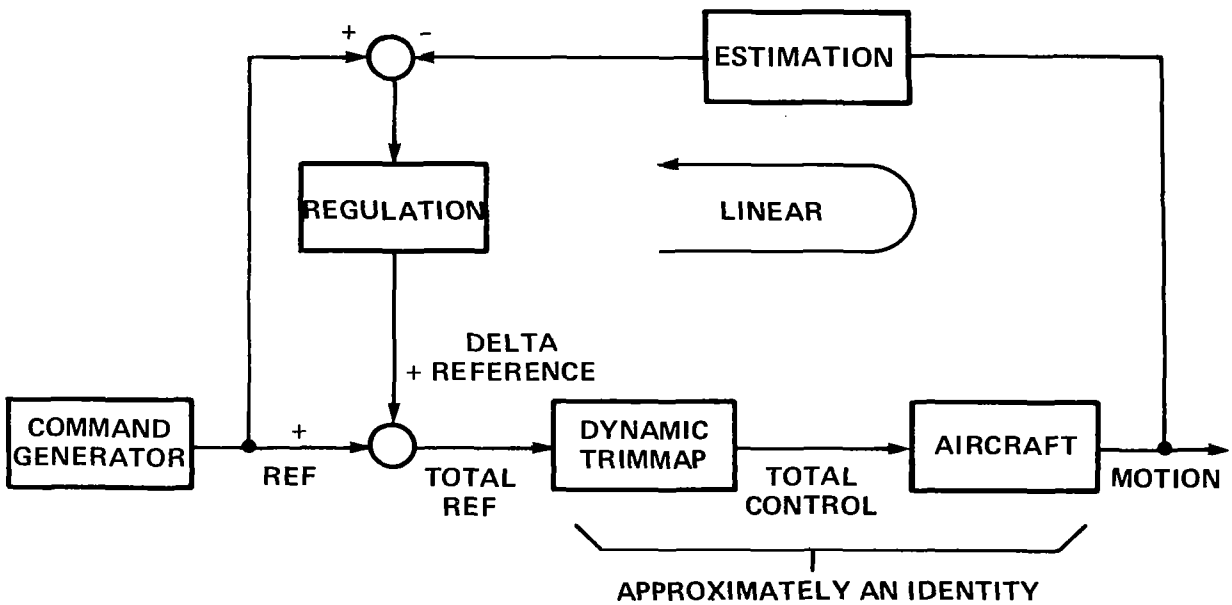


Figure 2.- Proposed arrangement of the loop control for TAFCOS.

"aircraft," with the control input as shown and the output the aircraft motion states. The blocks "command generator" and "autotrim" represent some feed-forward structure built into the controller design, and the block "regulation" represents some feedback logic.

In general, the "aircraft" block represents a vehicle with flight characteristics that are nonlinear. For STOL type aircraft, this nonlinear behavior can be quite strong and can be a major problem in the design of the controller logic. The use of gain scheduling, or the use of distinct flight modes, such as a cruise and a STOL mode, are possible solutions. The result is a potentially complex set of control logic with good performance characteristics only at certain discrete design points. The TAFCOS structure avoids these problems by modifying the structure to that shown in figure 2. The various blocks denote the same functions as before; however, the feedback point has now been moved to be in front of the trimmap. The trimmap concept is the heart of the TAFCOS design and provides the structure to handle the nonlinearities. By proper design of the trimmap, the control loop can be made to behave in a linear fashion over the entire operational range of the aircraft. The operation of the trimmap can be viewed as follows. Considering the aircraft as a point mass, the motion is then governed by the applied force, which in turn is proportional to the commanded acceleration. If a trajectory is defined for the aircraft to follow, the acceleration, and hence the required force, are then known. The trimmap represents an inverse model of the aircraft and for the commanded force solves for the control settings that will generate that force. If the trimmap contains an accurate representation of the vehicle characteristics, the controls generated from the trimmap when applied to the aircraft will result in a trajectory that duplicates the one commanded. From the standpoint of the control loop operation, the application of the trimmap concept makes the combined trimmap and aircraft blocks approximately an identity and results in operational characteristics that are approximately linear.

The structure of the trimmap, in greatly simplified form, is shown in figure 3. The data shown are a typical lift-drag plot for the Twin Otter aircraft. The complete trimmap is made up from data from several such graphs for the various parameters that model the aircraft. When the data shown on the graph are used to model the aircraft, alpha and throttle settings, when throttle is converted to C_T, are the inputs which specify the overall C_L and C_D for the aircraft. The actual computation for any aircraft is, of course, a good bit more complex as flap setting, propeller rpm, etc., have to be factored in as well. The forces computed are then used in the equations of motion to determine the resulting flightpath of the aircraft. To use these same data for the trimmap, the procedure is simply reversed. From the desired or commanded trajectory, specifically the acceleration for that trajectory, the forces required are determined. By an inverse procedure to that used in the modeling operation, the data are used to determine the alpha and throttle settings needed to generate those forces. The commanded alpha is then converted into control surface commands to drive the aircraft along the desired trajectory. Again, the operation can be complex because such factors as flap setting and engine rpm, etc., must be taken into consideration. The concept of the trimmap can be seen to be fairly straightforward, but it should also be obvious that in actual practice the configurational complexity of the aircraft can make its use computationally difficult. Any control redundancy must be resolved and procedures devised to efficiently sort through the data. The advantage in using the trimmap, however, is that as much detail as wanted can be built in without conceptual difficulty; this allows the combined trimmap-aircraft blocks to be an approximate identity. In addition, the trimmap will

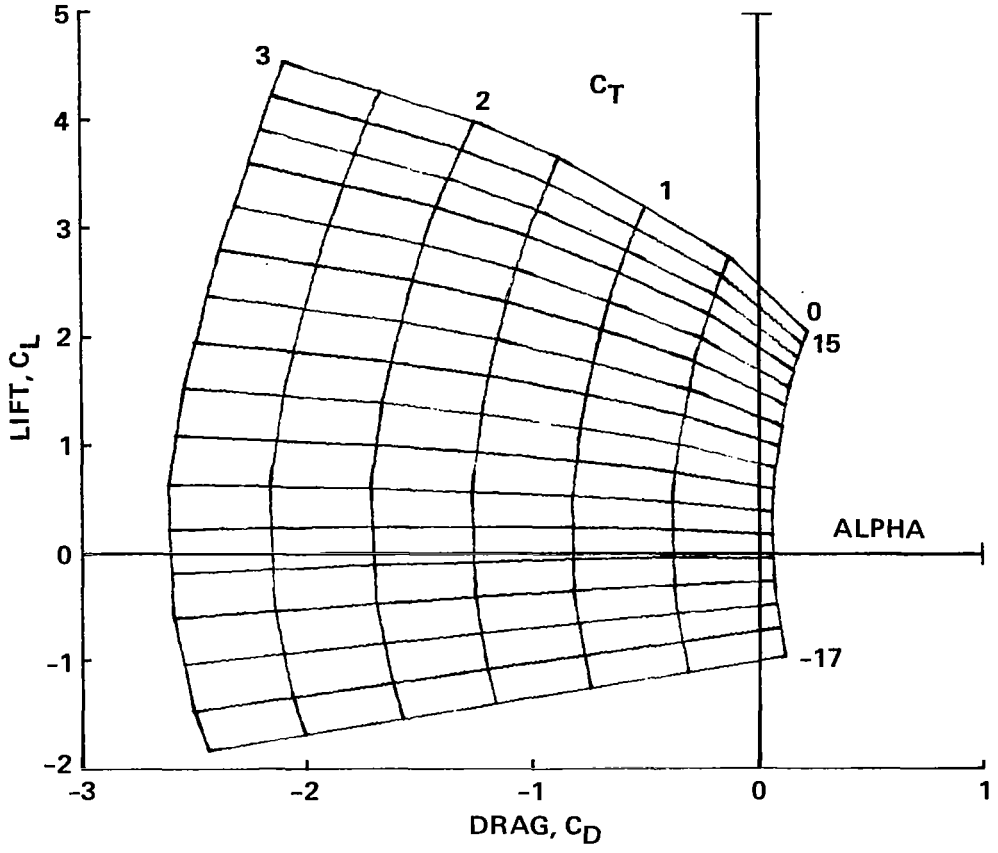


Figure 3.- Typical lift-drag polar for the Twin Otter aircraft.

permit the inclusion into the control logic structure of, for example, envelope limiting and control redundancy management.

To utilize the trimmap concept, TAFCOS has been structured as shown in figure 4. The TAFCOS structure is broken into the two main sections, as shown by the dashed lines in figure 4. These consist of two inner, or attitude control and power control, loops, and an outer, or trajectory, loop. Each loop has the structure described in figure 2 with a trimmap and feedback structure as shown. The attitude and power loops are fast loops relative to the trajectory loop, and in the mechanization for flight are operated at five times the speed of the outer loop. The inputs to the attitude loop are commanded attitude from the trajectory loop, measured aircraft attitude and angular rate, and the equivalent throttle information. No trajectory measurements or command inputs are provided to this section of the control logic. In like manner, the trajectory loop receives only the commanded trajectory and measured aircraft position, velocity, and acceleration. The trajectory loop has no knowledge of attitude or throttle variables. The result is a simplification of feedback structure with the elimination of coupling between the loops.

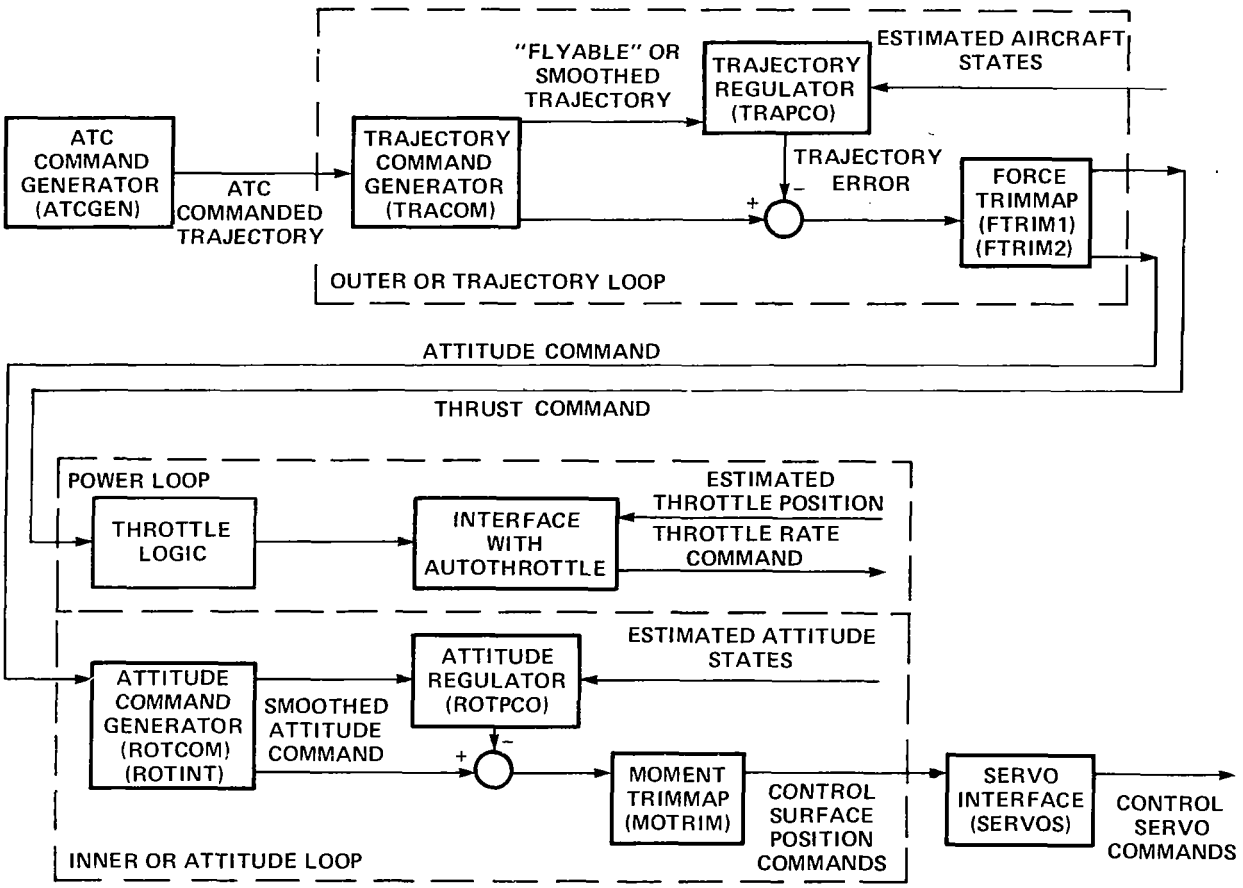


Figure 4.- TAFCOS structure.

The computational sequence for TAFCOS starts with the generation of the air traffic control (ATC) commands generated by the box at far left (fig. 4). Based on a stored set of way points, a sequence of straight line segments or circular arcs is generated for the aircraft to follow. These segments are not necessarily smooth and may have corners or possibly even disjointed sections. The command outputs are position, velocity, and acceleration as a function of time; they are given in the ground coordinate system. These signals can be treated by TAFCOS as either a three-dimensional or four dimensional command. With a three-dimensional command, the aircraft position is specified by the way points, but the velocity is controlled by the pilot or specified in terms of the airspeed profile to be flown; hence, the aircraft position along the commanded flightpath is not fixed in advance. With the four-dimensional command, the aircraft motion along the flightpath is fixed in terms of a specified ground speed.

The ATC command signals are the drive signals for the outer or trajectory loop. These signals are first processed by the trajectory command generator (TRACOM) which performs several functions. The output of the trajectory command generator is to be a smooth flyable trajectory, one that does not ask the

aircraft to perform maneuvers of which it is incapable. TRACOM is required to smooth any rough spots in the ATC trajectory, provide smooth transitions for any step commands, or smoothly reduce any large errors that may build up; in addition, it does any required limiting to commands so that they do not exceed aircraft capability. The output is again a position, velocity, and acceleration command in ground coordinates.

The acceleration output of TRACOM is the principal command signal for the aircraft trajectory control. This signal is proportional to the force required to perform the ATC maneuver which in turn is used as an input to the trimmap. Trajectory feedback information is provided at this point through the trajectory regulator (TRAPCO). The three output signals from TRACOM are used as an input to the regulator where they are combined with the states of the aircraft to form an error signal. The outputs of the perturbation controller are position error, velocity error, and the integral of acceleration error. These signals are summed with the acceleration output from TRACOM for an augmented acceleration signal for the trimmap computation. The trimmap computations (FTRIM1) then generate the required alpha and throttle settings to produce the desired force. The alpha command is converted to a commanded attitude for use by the inner loop computations.

The inner or attitude loop is essentially a duplicate of the trajectory loop except that moments and attitude commands are considered rather than force and trajectory commands. The output by the moment trimmap is the control surface setting required to generate the needed moments, and these control signals are then used to drive the aircraft control surfaces. Most of the structure of TAFCOS is derived from kinematic and dynamic principles and is therefore vehicle-independent. The trimmaps, of course, contain detailed information about the aircraft; in addition, TRACOM has a knowledge of the general limits of the aircraft flight capability. The result is that the basic structure of TAFCOS applies — with the construction of a suitable trimmap plus any modifications required to accommodate the different control configurations that might be encountered — to any vehicle. It should also be pointed out that TAFCOS does, in general, require the use of a digital computer to carry out the control operations. In appendix A, a detailed description is given of the contents of each of the blocks shown in figure 4.

TAFCOS FLIGHT IMPLEMENTATION

The flight evaluation of the TAFCOS concept was performed using a DHC-6 Twin Otter aircraft. The main reason the Otter was chosen is that it is equipped with a digital avionics system called STOLAND (ref. 3). The avionic system was designed for performing navigation, guidance, control, and display experiments on STOL research aircraft. STOLAND utilizes a Sperry-1819A digital flight computer as the central processor into which the TAFCOS control logic was programmed. Actual avionics flight hardware is also incorporated into a simulation facility which uses a digital computer for the aircraft modeling and has a fixed base cab for the display and piloting work. Figure 5 is a composite photograph of the various components that make up the simulator/flight-test setup.

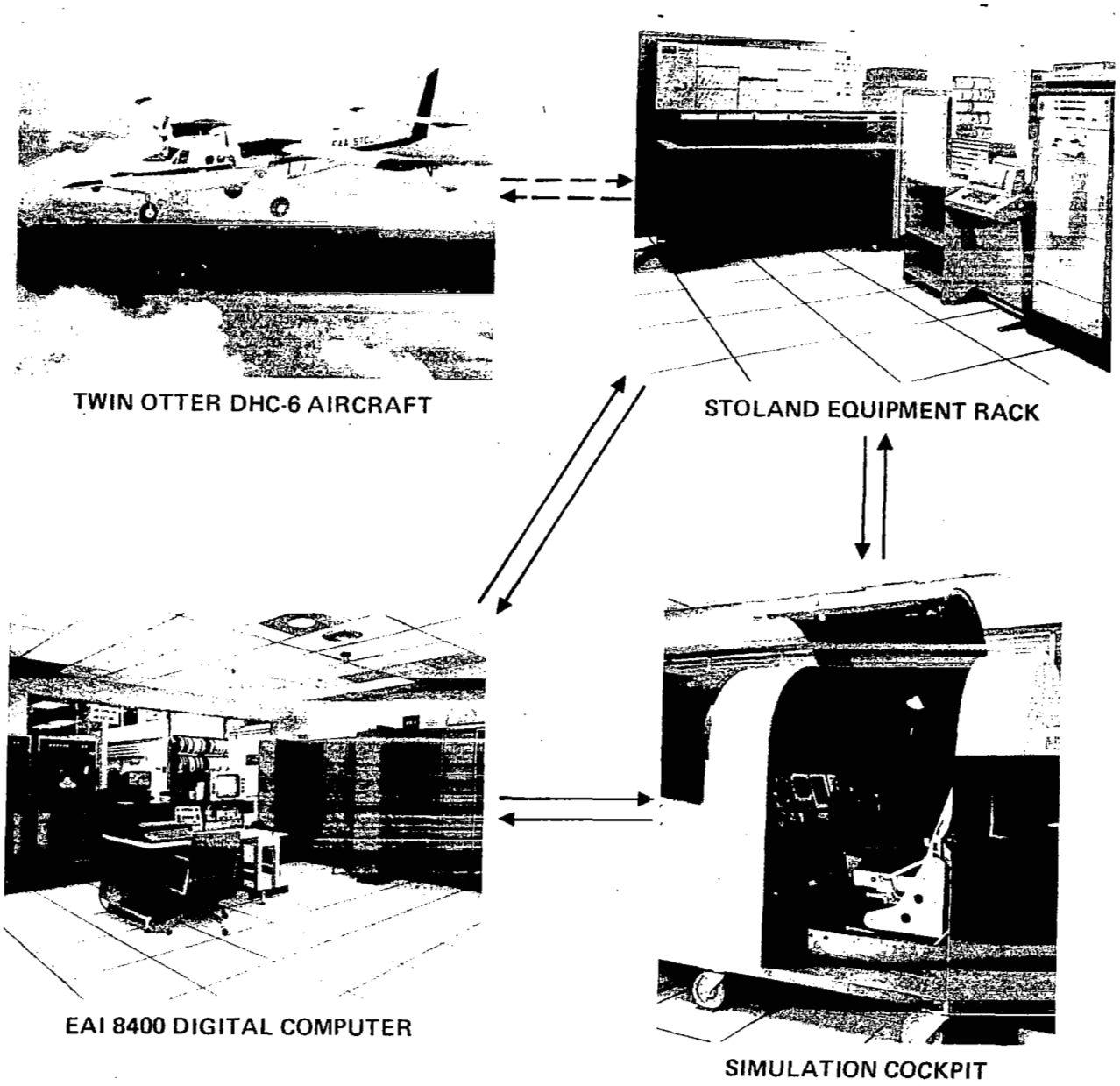


Figure 5.- Equipment used in the TAFCOS simulation/flight evaluation.

Both the cab and the aircraft have the usual complement of displays for pilot control and observation. In addition, there is an electronic flight director indicator, the EADI which is programmed through the 1819A, and a CRT moving map display, the MFD. A special mode select panel was included for control of the STOLAND system. The outputs shown on these instruments are controlled through the 1819A computer and are programmable by the engineer. A photograph of the cockpit installation is shown in figure 6.

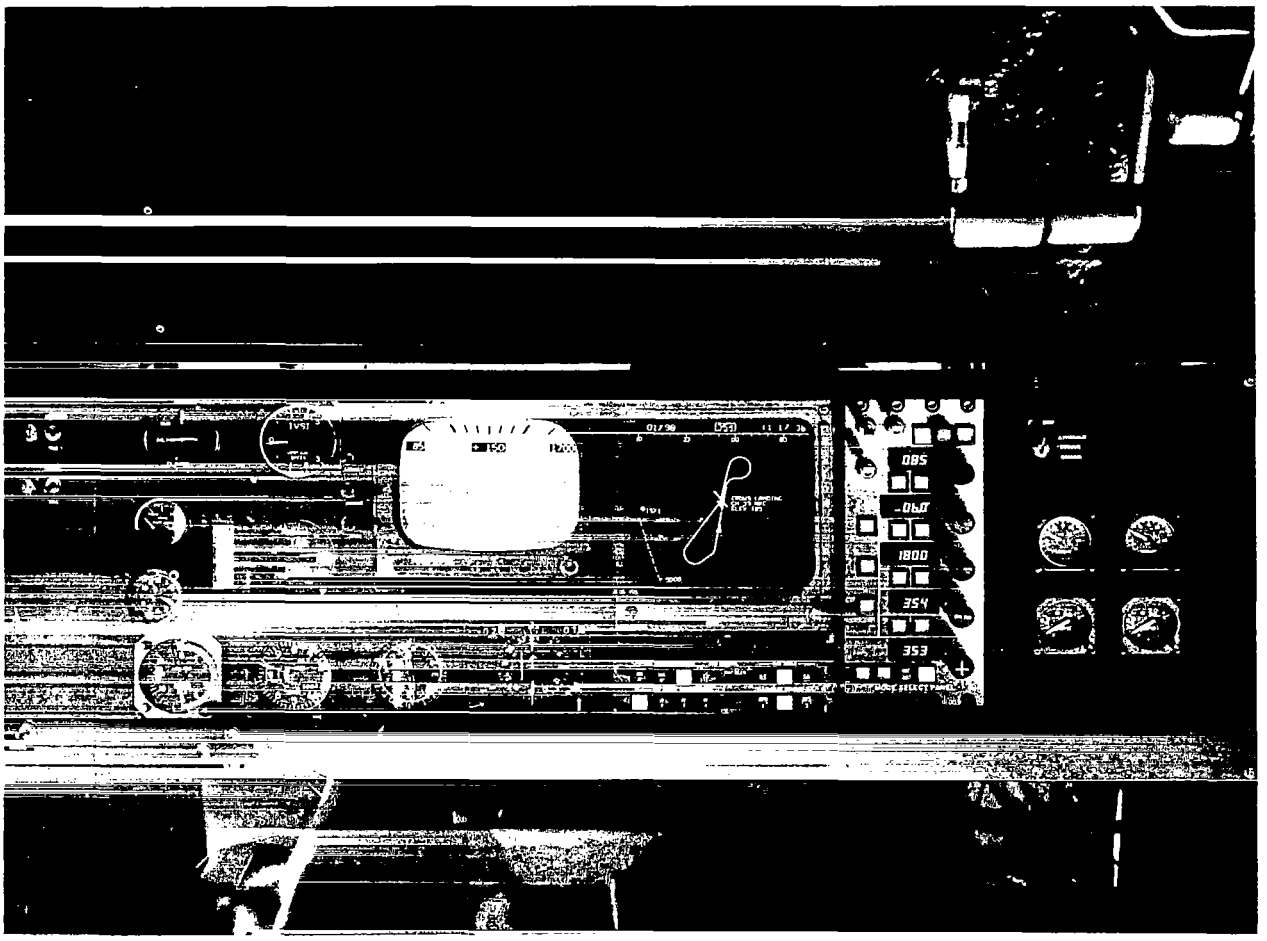
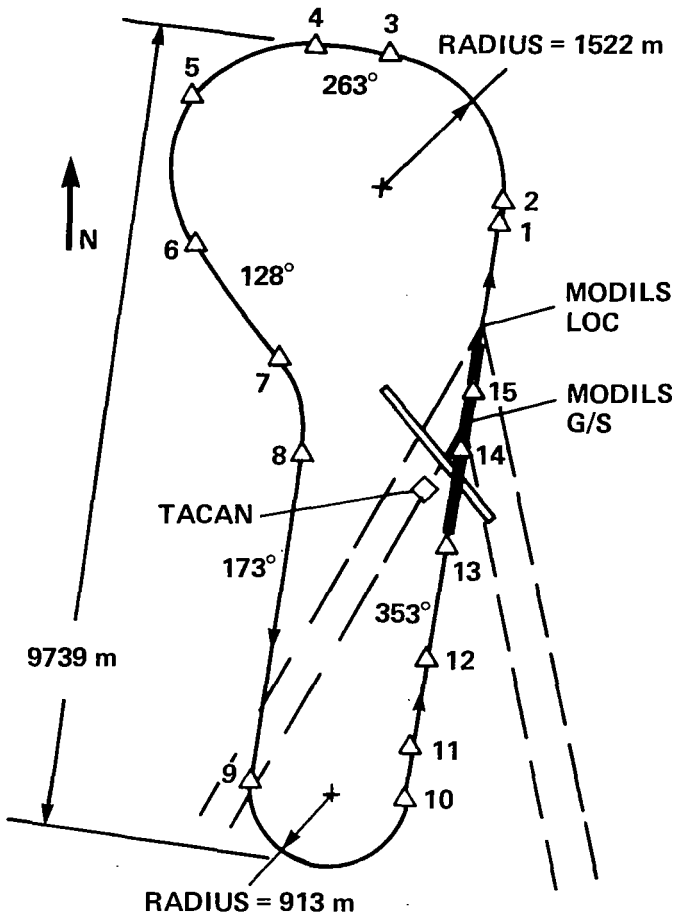


Figure 6.- Cockpit display equipment.

The STOLAND software package contains all of the items necessary for flight control of the aircraft, including a complementary filter-type navigator, display commands, guidance laws, and a variety of other features. The TAFCOS program was inserted in the STOLAND system by replacing all of the autopilot and SAS functions of the STOLAND guidance routine but retaining the navigation routine, the display functions, and all of the general housekeeping functions of the STOLAND operations. A more detailed description of the way in which TAFCOS and STOLAND were combined is given in appendix B.

FLIGHT PROCEDURES

For the flight evaluation, a trajectory was chosen that would exercise a good part of the flight envelope of the aircraft and hence evaluate TAFCOS over a wide range of flight conditions. The main features of the flight trajectory are shown in figure 7. To fly the path shown, initial capture was to



WAY POINT NUMBER	ALTITUDE* m	AIRSPEED m/sec
1	219	51
2	238	51
3	363	51
4	500	51
5	500	56
6	500	60
7	500	60
8	500	48
9	500	43
10	500	41
11	500	37
12	390	37
13	238	37
14	219	43
15	219	51

* ALTITUDE ABOVE SEA LEVEL
(FIELD LEVEL 42 m)

Figure 7.- Required trajectory for TAFCOS flight evaluation.

take place somewhere in the vicinity of way point 8, which is about midway along the downwind leg. The aircraft speed was to be about 51 m/sec and the capture altitude about 500 m above sea level (455 m above the runway). The aircraft was required to fly about 2-1/2 circuits around the pattern. For the repeated circuits, the aircraft made a normal descent toward the runway and leveled off at about 219 m for a go-around maneuver. The commanded path requires an altitude variation from 500 m to 219 m and requires airspeed variations from 60 m/sec to a low of 37 m/sec. The approach descent to the runway calls for a 6° glide slope and a 3° path for climb-out. A variety of flap settings was called for as well as changes in propeller rpm, with those changes made in accordance with standard flight procedures for the Twin Otter aircraft. TACAN was used as the navaid throughout most of the flight. The last approach to the runway was done with the use of a microwave landing system (MODILS).

There were two main objectives of the flight test, both of which were concerned with evaluating concept performance. The first objective was to demonstrate that the TAFCOS concept is practical for use as a flight controller. This was an objective because of the mathematical structure of

TAFCOS. As has been previously described, the TAFCOS concept is an integrated controller with sufficient structure to be capable of handling the difficult control problems of STOL and VTOL aircraft without the multimode operations or complex gain scheduling. The integrated structure of TAFCOS should, for a comparable level of control, provide for a more efficient and compact set of controller logic. However, the mathematical structure of TAFCOS is considerable (see the development in ref. 1), and the required mathematical operations could present a problem with the restrictions of space and time imposed by a typical flight computer. In the process of conducting a flight test, any difficulties in the generation of software would become immediately apparent, and it was considered important to show this not to be the case.

The second objective was to demonstrate that the TAFCOS concept could function well in a real-world environment, what might be called the robustness of the controller. A good deal of evaluation can be done in a simulation environment with the use of varying levels of wind and turbulence; however, the operation of TAFCOS depends on the use of a priori knowledge of the aircraft model and the actual aircraft may, and likely does, differ somewhat from the model used in the simulation. Flight evaluation of TAFCOS would verify that the structure of TAFCOS can handle these differences, that they will not cause a problem to the controller operation, and that, as a result, the expected level of performance can be maintained. A similar problem can be encountered with noise that exists on sensor and navigation signals. Data to evaluate these criteria would come primarily from the tracking capability of TAFCOS during the flight and from some monitoring of internal variables.

COMPUTER SPACE/TIME REQUIREMENTS

As just mentioned in the previous section, the practicality of the control logic of TAFCOS can, to a degree, be judged by the computer requirements needed to carry out the control tasks. A tabulation of the space-time requirements used with the flight-test program, the total available in the 1819A computer, and the amount used in the conventional set of control logic used by the basic STOLAND system are shown in table 1. The upper line on the chart shows the total memory available within the 1819A. The total cycle time of 50 msec is also shown as well as times required to carry out the add and multiply operations. The add and multiply times give an indication of the efficiency of the 1819A. Entries in the last column are measures of the available computational complexity; they are products of the space and time available (word-seconds). Those numbers are presented because with a digital program it is possible to tradeoff time and space by programming techniques. Similar data are shown for the control logic as programmed for the STOLAND system using conventional autopilot/SAS control laws and for the programming of the TAFCOS logic. Two sets of numbers are shown for TAFCOS. The upper set is for the programming as done for the flight test; those numbers are an actual count of the computer usage. For ease of operation, the TAFCOS programming was done with decimal scaling. However, the 1819A is a fixed-point machine that operates more efficiently with binary scaling. The last set of numbers is an estimate of the improvement that can be obtained by converting to binary scaling.

TABLE 1.- COMPUTER SPACE-TIME REQUIREMENTS FOR TAFCOS

	Computer memory locations, words	Time requirements, msec	Complexity, word-sec[a]
Sperry-1819A flight computer	32,768	Add: 2 Multiply: 24 Cycle time: 50	1640
STOLAND control strategy	8,237	12.5/50	103
TAFCOS			
Decimal scaling	2,523	9.3/50	24
Binary scaling (estimated)	2,000	5.0/50	10

[a]Product of computer memory locations used and computation time.

The numbers shown in table 1 indicate quite clearly that the integrated structure of TAFCOS does provide for a more compact set of control laws than was the case for the conventional approach. It is recognized that a comparison of this sort may not be completely fair to the conventional set of logic as no real attempt may have been made to generate a program that made optimal use of computer memory and time. The difference is sufficiently large, however, that it would seem clear that the TAFCOS approach does have an inherent advantage over the conventional logic. The main reason for the difference appears to be that TAFCOS does not require the separation of axes and modes that normally are used in conventional designs that call for a duplication of logic and, therefore, a larger program.

A question that can be asked on the TAFCOS programming is how the complexity of the software would change with a different vehicle. As has been mentioned, TAFCOS was conceived as a controller structure suitable for complex STOL and VTOL vehicles and clearly the Twin Otter is not in that category. The following list shows the breakdown of the TAFCOS programming.

TAFCOS Computational Section	Memory Locations Used
ATC command generator (ATCGEN)	480
Trajectory loop (force trimmap: 197)	902
Attitude loop (attitude trimmap: 163)	506
Servos	118
Data input	203
Miscellaneous housekeeping	314
Total	2,523

For the way in which TAFCOS is constructed, most of the computations are not vehicle-dependent and are constructed on basic kinematic principles. Some computations do depend on vehicle characteristics, but they are mostly limited to the setting of various gains and limits. Only the trimmap sections, which

are inverse models of the aircraft characteristics, are truly vehicle-dependent. Therefore, a change to a more complex vehicle would entail only a change in the size of the trimmap sections. The trimmaps for the Twin Otter were set up using an analytic description of the aircraft characteristics by curve fitting to the data charts for the aircraft; they required storage of about 38 numbers and storage of the computational instructions. The equations can be seen in the appropriate sections in appendix A. Trimmaps for complex vehicles might be considerably larger. As an example, for the Ames Augmentor Wing aircraft (a STOL vehicle with internally blown flaps, movable thrust nozzles, etc.), the tabular trimmap would add about 800 memory locations to the program. Considering the computational power of TAFCOS and the nature of the aircraft being controlled, the benefits of the integrated structure of TAFCOS are clear.

FLIGHT RESULTS

The first set of flight data presents an overview of the performance of TAFCOS during the flight test. Shown in figure 8 is a tracking of the aircraft by the ground radar system that is used in conjunction with the data collection system at the flight-test facility. The plots on the figure show an X-Z plot and an X-Y plot of the radar data. The X-axis represents distance along the runway center line, the Y-axis distance perpendicular to the runway, and the Z-axis is altitude. Note the expanded scale on the altitude variable.

As previously mentioned, the pattern was flown in a counterclockwise direction with the initial start point as noted. Because the aircraft flew approximately 2-1/2 circuits around the field, the tracks are repeated. At the initial capture point, the aircraft was flying at an airspeed of about 62 m/sec and was commanded to capture a path that was the extension of the downwind leg parallel to the runway. The lateral error at this capture was quite large, initially of the order of 600 m, and the vertical error was about 25 m (these numbers are actually somewhat larger than those shown in the figure because TAFCOS was in operation prior to radar lock-on). The flightpath terminated on an approach to the runway.

Figure 8 shows that TAFCOS did a good job of guiding the aircraft around the racetrack pattern with reasonably smooth performance. The scale of the figure does not, of course, permit much detail to be shown, but it is clear that the basic operation of TAFCOS as a flightpath tracking controller was satisfactory. The variations in the path guidance that are evident on the plot are from one of two main sources. The noise on the vertical axis is primarily due to turbulence. There was little or no ground wind during the flight, but the turbulence was rated as moderate; the result was that the vertical tracking was not smooth. The variations on the X-Y plot are due for the most part to variations in the TACAN signal with a resultant wandering of the navigation information. A significant distortion of the flightpath can be seen as the aircraft flew down the runway. At that location, the aircraft went through the cone of confusion for the TACAN station, which resulted in a large transient in the navigation data inputs.

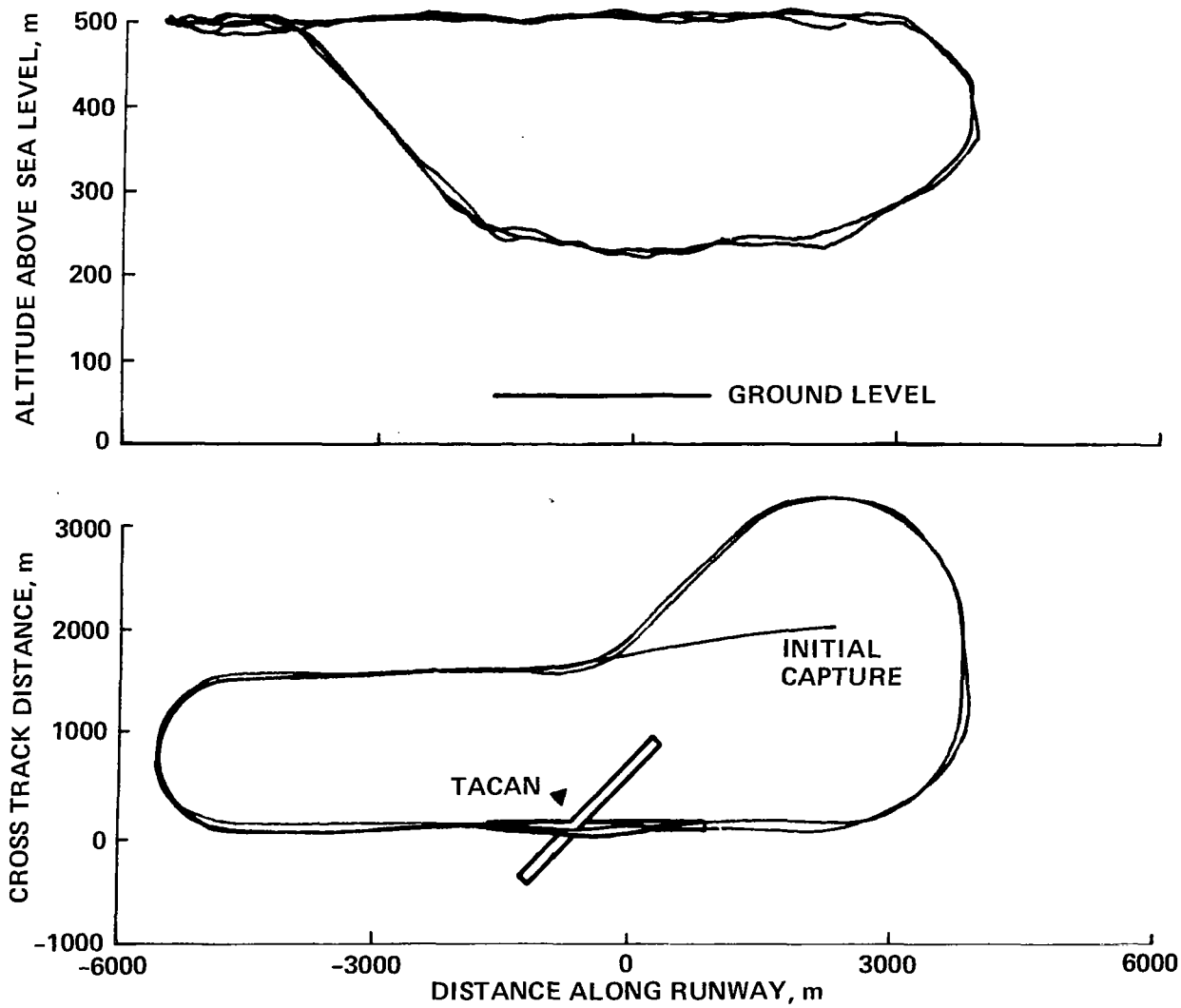


Figure 8.- Radar track of the TAFCOS flight test.

Of particular interest in figure 8 is the performance of TAFCOS during the initial capture. The lateral error was large and presented a situation in which stability problems could arise if the controller had not been properly constructed. The section of the logic called TRACOM was designed to handle such large errors when the output command to the remaining TAFCOS logic is a flyable command, even with such large input errors. It is clear from figure 8 that no difficulties occurred and the large error was eliminated in a well-controlled manner.

The next set of figures, in which the lateral and vertical deviations from the commanded path are shown, presents a more direct assessment of the performance of TAFCOS for the flightpath tracking task. In figure 9, the deviation data for the complete 2-1/2 circuits are given. The commanded

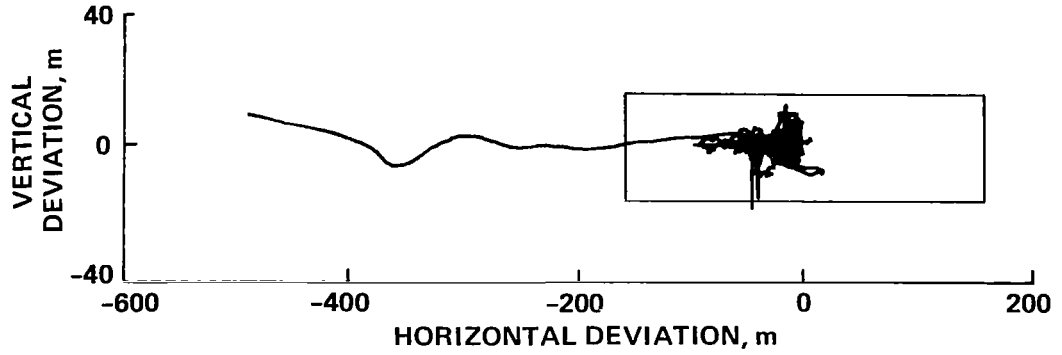


Figure 9.- Lateral and vertical deviations — TAFCOS flight controller operating over the full ATC trajectory.

position used in the generation of the data is the trajectory commanded by the ATC command generator, and the aircraft position is taken from the output of the navigation routine, the estimated aircraft position.

The format for the presentation is similar to that in which the information was presented to the pilot on the EADI. When the aircraft was under the control of the TAFCOS logic, a rectangular box appeared on the CRT to provide information on the location of the commanded flightpath relative to the aircraft. The box can be considered to be a tube through which the aircraft is to fly; therefore, it gives a qualitative visual measure of aircraft performance. The box is shown on the figures for essentially the same reason. In addition, the scaling of the plot is the same as that used on the EADI display.

A look at the variation of the data in figure 9 shows that the TAFCOS controller provided a level of tracking ability that was held to about ±12 m vertically and about ±75 m horizontally. The initial capture path is shown by the long line tailing off to the left. Essentially, all the data shown are for times when the aircraft was using TACAN for the navigation information to the lateral axis. Considering the level of turbulence that was present during the flight, the control of the vertical axis appears to be quite good, with the limits within the expected accuracy level of the altimeter. The lateral axis shows considerably more deviation, and examination of other data has shown that most of the scatter appears to come from variations in the estimated position of the aircraft by the on-board navigator; in turn, those variations were due to TACAN signal variations. Comparisons between the position of the aircraft as measured by the radar and the on-board estimator shows variations of the order of ±180 m, with relatively rapid changes in some regions. TAFCOS saw those variations as inconsistencies with other measurements, that is, velocity and acceleration, and developed hang-off errors to compensate. Despite those problems, the data show that TAFCOS was able to track the desired flightpath with reasonable accuracy and to do so under the required variety of flight conditions and changes in aircraft configuration.

Figure 10 shows a similar set of deviation data but only for that portion of the flightpath where the navigation routine had switched to the microwave landing system. The form and scaling of the data shown on this figure are the same as presented in figure 9. TAFCOS is unchanged, no mode or gain changes were made with the switch to the microwave landing system, and the only difference is the improved navigation signal. There is, of course, a very dramatic improvement in the performance of TAFCOS with the higher quality navigation information. The vertical deviations are now limited to about ±2.5 m and the lateral deviations to about ±10 m. It would appear from figures 9 and 10 that TAFCOS was operating up to the quality of the navigation signal.

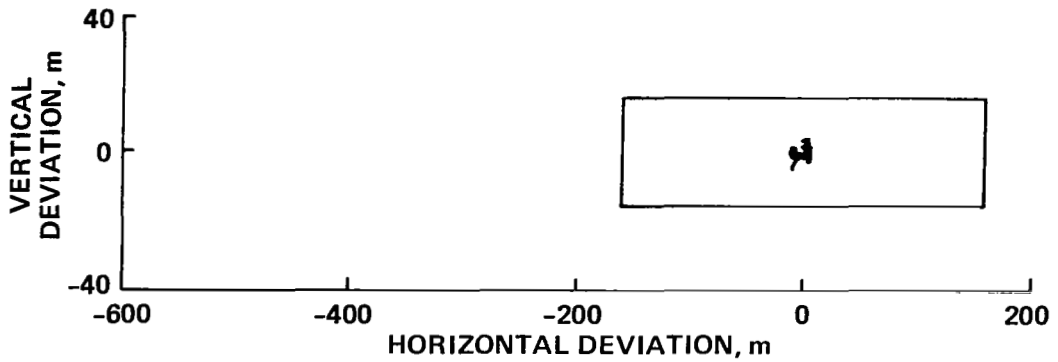


Figure 10.- Lateral and vertical deviations during final approach — TAFCOS flight controller with the microwave landing system.

The next set of data shows a somewhat different view of the TAFCOS performance. In figure 11, a comparison of the acceleration commanded within TAFCOS and the vehicle response provides a measure of whether the internal structure is in fact functioning as intended. Shown in figure 11 are time histories of accelerations; the upper curve of each pair is the commanded value and the lower the measured acceleration from accelerometers on board the single aircraft. The data are shown for the time history of a single pass around the field.

To interpret the data shown, it should be remembered that the principal drive signal is the commanded acceleration from the block called TRACOM of the trajectory loop. TAFCOS can, therefore, be considered as an acceleration controller, and the response of the vehicle should be a reasonable mirror of this drive signal. The data in figure 11 show that the commanded and measured values are a good match. There is, of course, a good deal of noise on the measured values, especially on the vertical axis; it is principally due to the turbulence encountered during the flight. A partial set of data from another flight, when the turbulence was low, shows much cleaner data traces. Despite the noise, however, it can clearly be seen that the vehicle is tracking the commanded accelerations. Key acceleration events can be seen by noting that way point 9 is the beginning of the turn from the downwind leg to the base leg. Then, after flying along the runway, the aircraft begins a series of turns, back onto the base leg (see fig. 7), that are reflected in a rather complex

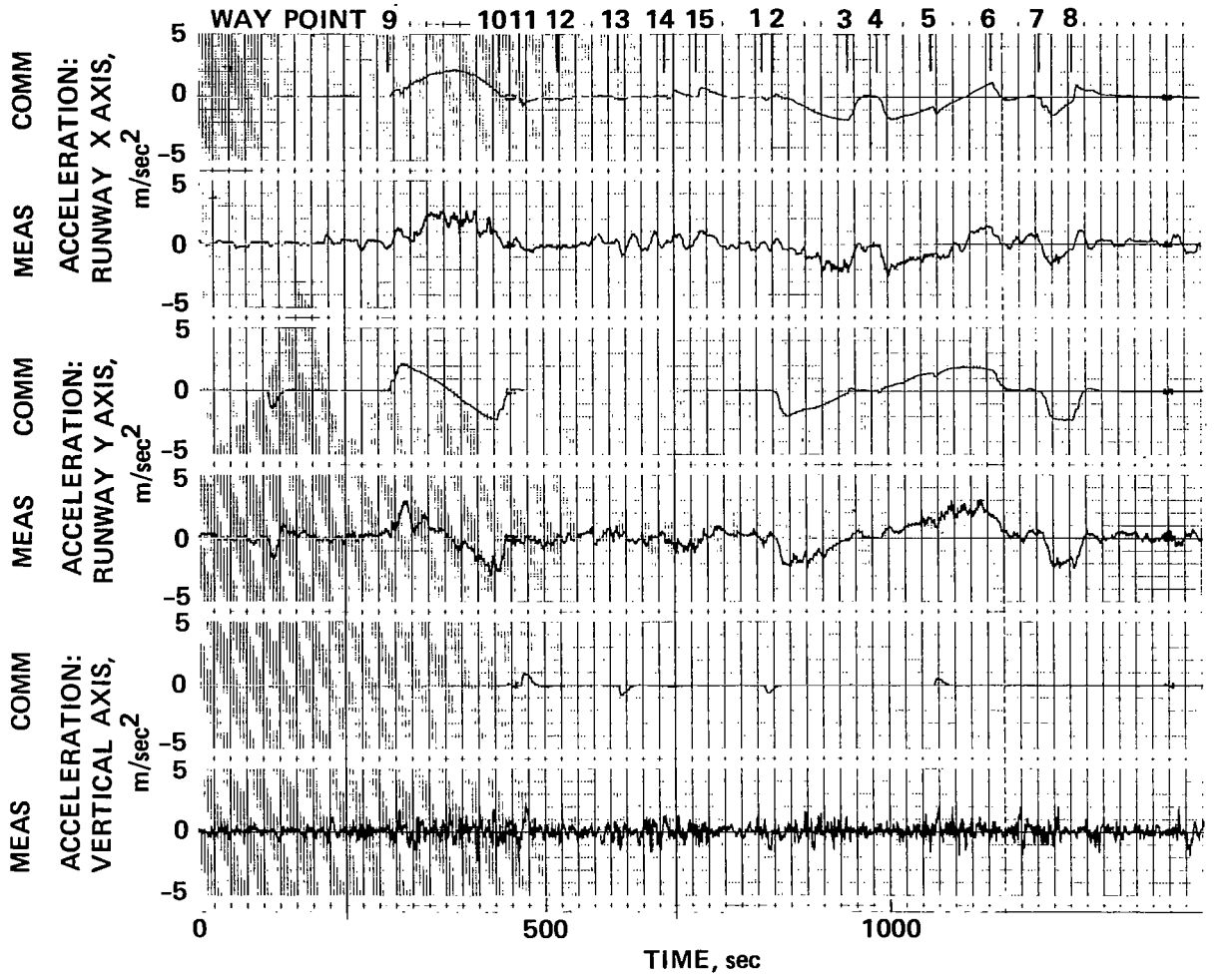


Figure 11.- Commanded and measured accelerations.

set of acceleration commands. In all cases, it can be seen that the measured values follow the commands; this shows that the kinematic and dynamic structures of TAFCOS, and the trimmap concept, are functioning as expected.

The next figure shows a comparison of TAFCOS deviation data with similar data for a conventional-type controller. In the beginning of this section, the programming size of TAFCOS was compared with the programming for the STOLAND guidance. A performance comparison between the two, to determine if the same level of performance is maintained with the more compact structure, is indicated. As mentioned previously, TAFCOS was inserted into the STOLAND software by replacing the autopilot and SAS functions built into the STOLAND system. The programming for these autopilot and SAS operations was not in fact removed but only bypassed on command. The result is that it was possible to simply switch from one set of control logic to the other during the flight. After the completion of the 2-1/2 passes around the field under TAFCOS control,

the pilot flew the aircraft back to the way point 8 capture point and flew another pass to about midway down the 6° final approach path using the conventional guidance provided by STOLAND. The STOLAND guidance used the reference flightpath mode, which is operationally identical to that used by TAFCOS. The data presented in figure 12 are the lateral and vertical deviations for that last approach and for TAFCOS flying the same portion of the trajectory. TACAN was used as the navigation source for the downwind and base-leg portion of the flight, with a switch to the microwave landing system for the final approach (the switch shows in the data as the bulge to the left on both plots). The flight conditions for both sets of data are nearly identical. The TAFCOS run was completed first; the time between the end of the TAFCOS run and the start of the STOLAND run was about 2 min.

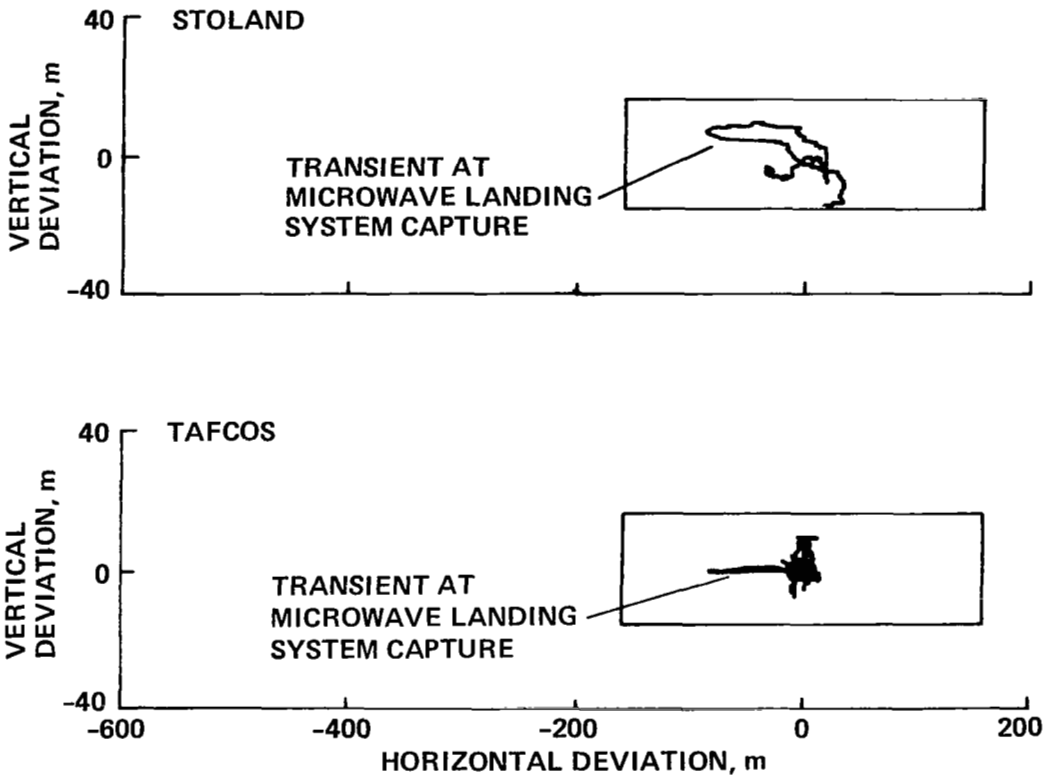


Figure 12.- Comparison of TAFCOS and STOLAND controller operation.

The data in figure 12 show that the performance of TAFCOS was equal to or better than that of the STOLAND guidance logic. The reduction in computer requirements for TAFCOS appears to cause no reduction in performance level and to give slightly better control. With more experience in the use of TAFCOS, improved modeling could provide a better a priori estimate for the trimmap computations, and the performance of TAFCOS should show further improvement.

CONCLUDING REMARKS

Flight-test results for the TAFCOS controller have verified that the theoretical development behind TAFCOS is valid and produces a practical control system. The integrated nature of the control logic results in a system that is capable of providing good flight control over a wide range of the aircraft flight envelope without the need for gain programming or special mode-select options. This performance was realized with very efficient computer space and time usage and showed that TAFCOS is practical for use in a typical digital flight computer. Work is in progress at Ames Research Center to flight-test a TAFCOS-type controller on the Augmentor Wing aircraft. The flight tests on the DHC-6 Twin Otter aircraft demonstrated the overall performance of the controller concept but did not require full utilization of the ability of TAFCOS to handle vehicles with highly nonlinear characteristics. The test with the Augmentor Wing should provide just such a data base.

In another program that is just beginning, the TAFCOS concept will be applied to the UH-1H helicopter. Because the structure of TAFCOS is mainly built on kinematic and dynamic principles, it is relatively vehicle-independent; consequently, it should be readily adaptable to this new problem and provide a workable solution to a difficult control problem. In a related task, the inclusion in TAFCOS of a manual control mode was investigated. Reference 4 presents simulation results on this work and demonstrates that TAFCOS is suitable for use with conventional autopilot modes. Also, reference 5 reports a simulation study using TAFCOS for carrier landing guidance.

Ames Research Center
National Aeronautics and Space Administration
Moffett Field, California 94035, September 20, 1979

APPENDIX A

TAFCOS STRUCTURE

A complete theoretical description of the TAFCOS structure, including the mathematical basis for the controller, and a simulation application of the concept to the Ames Augmentor Wing STOL aircraft are presented in reference 1. Flight-testing the TAFCOS concept on the Twin Otter aircraft required a number of changes in its structure. Some changes were required because of the particular aircraft used and because of the need for compactness in the final program due to the limitation imposed by the size and speed of the flight computer. This appendix describes the mathematical structure for TAFCOS that was used in the flight test, with particular emphasis on the build-up of the equations in the flight program. A program listing is given in reference 6.

The variable labeling used in the sections that follow is, to some extent, a mixture. When equations are discussed, the symbols are of the usual subscripted-type notation, with the coordinate frame and type of variable given by the subscripts. However, some of the computations are given in terms of block diagrams, and some internal variables are noted in terms of the computer symbolism rather than the subscripted type. These variables are not generally discussed and are shown on the diagram only for ease in following the program listing; their presence should present no confusion to the reader.

ATC and Trajectory Loop Operations

The computational sections that comprise the slow loop are those shown in figure 4 as the "air traffic control" (ATC) command inputs and the three sections shown in the box marked "trajectory" or "outer loop." As has been discussed in the main text, the output of the ATC section is the basic trajectory that the aircraft is required to follow and is given as a time history of position, velocity, and acceleration for that path. In an idealized sense, the trajectory loop then takes the commanded acceleration, which can be considered as a commanded force, and outputs the required vehicle attitude and setting of the throttle handle that will generate the necessary force. Vehicle configuration commands, such as flap setting and propeller rpm, could also be output commands if the vehicle had servo operation of these items; otherwise, these become measured quantities used to specify the other variables. Measured position, velocity, and acceleration are also used as inputs as a means of providing feedback control. The flight program computations carry out these operations in a series of five steps for an update of the output every five computer cycles. The sections that carry out the operations are labeled ATCGEN, COMVA/TRAPCO, TRACOM, FTRIM1, and FTRIM2. The block diagram in figure 13 shows how the blocks are connected and the variables that are transferred between them.

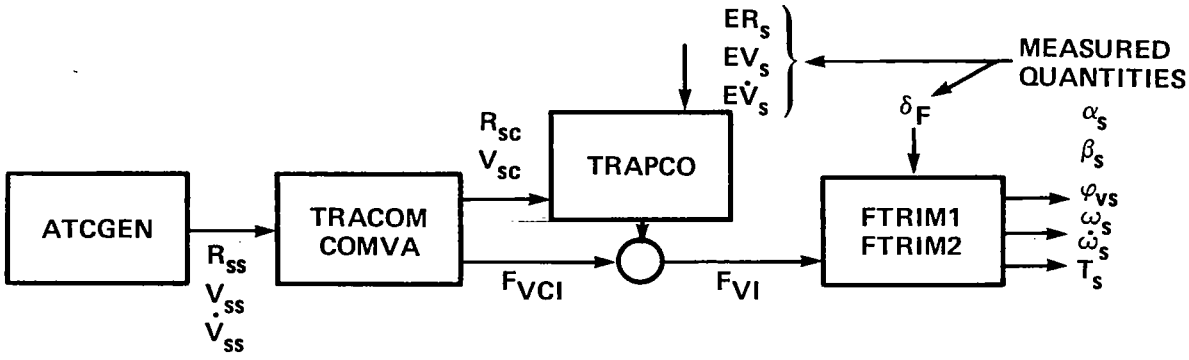
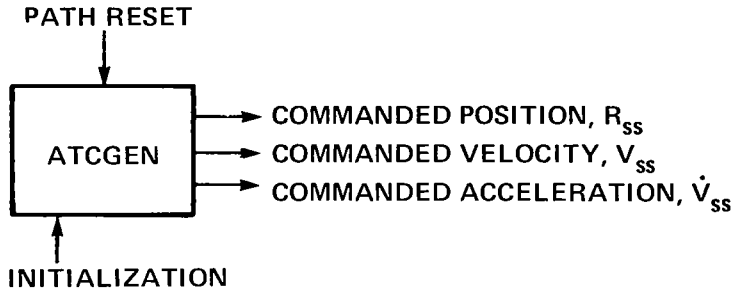


Figure 13.- Block diagram of ATCGEN and trajectory loop interconnections.

ATCGEN

The purpose of ATCGEN is to generate the trajectory commands for TAFCOS to follow. ATCGEN is not properly a part of the TAFCOS controller. It has been assumed that a fairly sophisticated ground/air ATC control system, development of which would be a major task in itself, would be needed for actual use in commercial service. However, in the absence of such a system, it was necessary to construct a relatively simple one for the current work.

The input/output for ATCGEN is shown below. The commands generated by



ATCGEN are essentially open loop. The computations must be initialized at the time of turn-on so that the ATC computations know the aircraft position and the starting way point for the flightpath to be flown. The information for the desired flightpath is stored in the computer as a set of way points and the task for ATCGEN is to take those points and construct a flightpath that consists of a series of straight-line segments or helical arcs. In addition to the position information, the way-point data also specify the airspeed at each location.

The method of computation in ATCGEN is based on a knowledge of path length s between way points and of how far the aircraft has traveled along that path. With a knowledge of the path shape, either straight line or helix, the position, velocity, and acceleration commands for the aircraft can be

computed. Defining a unit vector tangent to the commanded path, the following set of equations can be used to define the path commands.

At any point on the commanded path, the unit tangent vector can be shown to be of the form shown below where ψ_c and γ_c are the commanded values of heading and glide slope for that particular path.

$$u_s = \begin{pmatrix} \cos \psi_c \cos \gamma_c \\ \sin \psi_c \cos \gamma_c \\ -\sin \gamma_c \end{pmatrix} \tag{A1}$$

The values of ψ_c and γ_c can be obtained from the components of the commanded velocity vector by the following equations:

$$\left.\begin{aligned} \begin{vmatrix} \psi_c \\ \gamma_c \end{vmatrix} &= \text{stored values at way point} \quad \text{(for a line)} \\ \psi_c &= S_{ATC} \frac{\cos \gamma_c}{R} \\ \gamma_c &= \text{stored value} \end{aligned} \right\} \quad \text{(for a circle)} \tag{A2}$$

If the path to be flown is a straight line, the values are fixed and stored in memory. A curved path will call for a continuous update from the present value of the commanded velocity. The path derivative of the unit vector given by equation (A1) will be needed in later computations; those quantities are given below:

$$\left.\begin{aligned} \frac{du_s}{ds} &= 0 \quad \text{(for a line)} \\ \frac{du_s}{ds} &= \begin{pmatrix} -\sin \psi_c \cos \gamma_c \\ \cos \psi_c \cos \gamma_c \\ 0 \end{pmatrix} \frac{\cos \gamma_c}{R} \quad \text{(for a helix)} \end{aligned} \right\} \tag{A3}$$

where R is the radius of the circle in the ground projection of the path.

The position command R_{SS} can now be computed from the equation below. The computation is in terms of the path length variable S_{ATC} which is the distance from the way point behind the aircraft.

$$R_{SS} = R_{SS}(0) + S_{ATC}(u_s) \quad \text{(for a line)} \tag{A4}$$

$$R_{SS} = R_{SS}(0) + \begin{pmatrix} R \sin \psi_c \\ -R \cos \psi_c \\ S_{ATC}[u_s(3)] \end{pmatrix} \quad \text{(for a helix)} \tag{A5}$$

In a like manner, the velocity commanded by ATC can be computed where the quantity V_{ATC} is the velocity associated with the point specified by S_{ATC}:

$$V_{SS} = V_{ATC}(u_s) \tag{A6}$$

The acceleration command associated with the above position and velocity commands must be handled in a somewhat different way because the acceleration is due to a combination of the effects of path curvature and the change in velocity along the path. The curvature term is given by:

$$\dot{V}'_{SS} = (V_{ATC})^2 \left(\frac{du_s}{ds}\right) \tag{A7}$$

Note that the derivative of the tangent vector is proportional to 1/R (see eq. (A3)); therefore, the quantity shown is simply the centripetal force acceleration for the circular arc paths. The other part of the acceleration comes from commanded velocity change and is handled in a manner similar to a servo error. A velocity error is calculated by solving the difference between the required velocity at the next way point and the present true airspeed of the aircraft. This difference is multiplied by a gain term and is limited to a preset maximum acceleration to construct the acceleration command:

$$\dot{V}_{ATC} = K_{VATC}[V_{ATC}(NWP) - V_T] \Big|_{\substack{\text{limited to} \\ \pm\dot{V}_{ATCL}}} \tag{A8}$$

The total acceleration command is then the sum of equations (A7) and (A8).

$$\dot{V}_{SS} = \dot{V}'_{SS} + \dot{V}_{ATC} \tag{A9}$$

The computation thus far has assumed a known path length position and associated velocity, S_{ATC} and V_{ATC}, and these must, of course, be computed for use in the equations given. The equations used are given below.

$$\left.\begin{aligned} V_{ATC} &= V_{ATC}(0) + \dot{V}_{ATC}T \\ S_{ATC} &= S_{ATC}(0) + V_{ATC}T \end{aligned}\right\} \tag{A10}$$

where T is the outer loop sampling time of 250 msec. S_{ATC} must also be adjusted for the effects of winds if the aircraft is flying under a three-dimensional plus airspeed mode. Computationally, this means simply resetting S_{ATC} as appropriate to compensate for the fact that S_{ATC} is a ground or

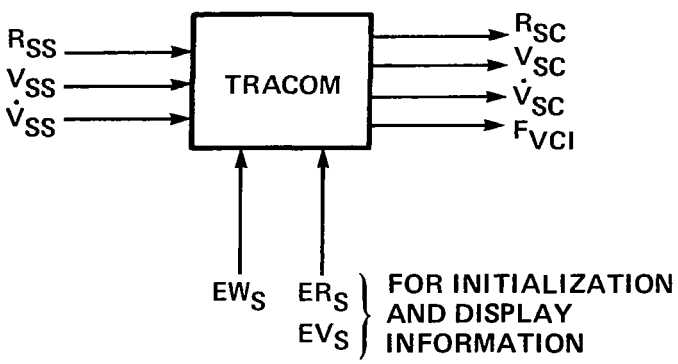
inertial quantity and the velocity term is an airspeed quantity. An inconsistency will develop between the two if there are winds. The computation of the reset value is done in a later section and will be discussed then.

A last item in the trajectory generation discussion is the problem of segment switching. At a way point, in general, there is some command change from the stored way-point information. If there is a change in heading or glide slope, there will be a transient introduced because of the "corner." To smooth out those transitions, ATCGEN switches early from one segment to the next. When the aircraft approaches a way point, at a path length equal to the distance the aircraft will travel in 4 sec, the ATCGEN logic switches to the new segment. An analysis has shown that 4 sec provides for smooth transition at all airspeeds.

One other function of ATCGEN is to command the flap setting required for the speed profile to be flown. The flap settings are stored as way-point data with values at 10° increments from 0° to 40°. Computations are also performed to impose airspeed and flap-angle limitations so that the aircraft remains within the placard requirements.

TRACOM

TRACOM provides the smoothing required to take the ATCGEN commands, the acceleration, velocity, and position time histories, and to convert those signals into a flyable set of trajectories for the aircraft to follow. The problem presented by the ATCGEN trajectories is that they may have discontinuities, such as "corners" or jumps in one or more of the variables at way-point changes, and at initial capture of a reference flightpath the deviations from the path would quite likely be very large. In order to avoid saturation of signals within the main structure of TAFCOS, which could potentially result in stability problems, TRACOM shapes the input commands to be always smooth and flyable so that the approximate linearity of the TAFCOS controller is preserved. As mentioned in the ATCGEN section, the inputs to TRACOM are the trajectory commands R_{SS}, V_{SS}, and $\dot{V}_{SS}$. The outputs are an equivalent set of trajectory commands R_{SC}, V_{SC}, $\dot{V}_{SC}$, and F_{VCI}. The last term of the output commands is the commanded specific force to be generated by the force generation process of the aircraft. A block diagram of the input/output variables is shown below.



Before discussing the structure of TRACOM, it is convenient to first carry out a support calculation that generates a direction cosine matrix required at a number of locations throughout TRACOM and other parts of the TAFCOS logic. In general, most of the TAFCOS computations are performed with the variables defined in one of three reference frames. Trajectory commands are given in terms of inertial or ground coordinates. Computations that involve aerodynamic terms are best defined in terms of a velocity coordinate frame tied to the airspeed vector or in the body frame. The matrix used to transform variables from the inertial to the velocity frame is generated in a section of the flight program called COMVA. Defining the commanded airspeed by a following equation, where V_{SC} is the commanded airspeed in the ground frame and EW_S, is the wind estimate from the navigation system,

$$\left.\begin{aligned} EV_{RWSC} &= V_{SC} - EW_s \\ ||EV_{RWSC}|| &= EV_c \\ EVSC &= \frac{EV_{RWSC}}{EV_c} \end{aligned}\right\} \tag{A11}$$

and in terms of heading and glide slope through the matrix A_{VSC},

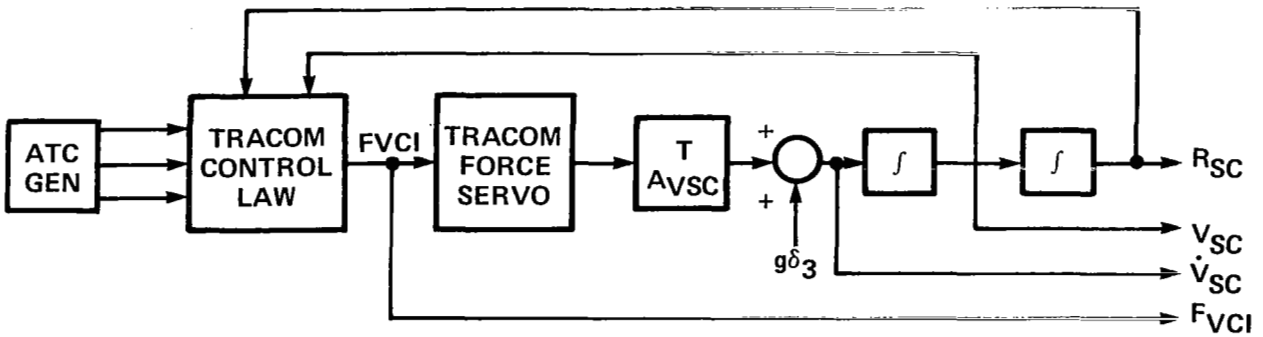
$$(u_i) = E_{VSC} = A_{VSC}^{t}\delta_1 \tag{A12}$$

The direction cosine matrix between the ground frame and the commanded velocity frame can now be constructed with the assumption that the X axis of the velocity frame is aligned with the velocity vector and the Y axis is horizontal.

$$A_{vsc} = \begin{pmatrix} t_1 & t_4 & t_7 \\ t_2 & t_5 & t_8 \\ t_3 & t_6 & t_9 \end{pmatrix} = \begin{pmatrix} u_1 & -t_2/t_9 & -t_5/t_3 \\ u_2 & t_1/t_9 & t_4 \\ u_3 & 0 & \sqrt{1-u_s^2} \end{pmatrix}$$

$$= \begin{pmatrix} \cos\gamma_v \cos\psi_v & \cos\gamma_v \sin\psi_v & -\sin\gamma_v \\ -\sin\psi_v & \cos\psi_v & 0 \\ \sin\gamma_v \cos\psi_v & \sin\gamma_v \sin\psi_v & \cos\gamma_v \end{pmatrix} \tag{A13}$$

With the completion of the matrix definition, it is now possible to go into the main computations performed by TRACOM. Conceptually, TRACOM can be considered to be structured as shown on the following diagram:



In a sense, TRACOM contains a relatively coarse model of the aircraft so that input commands from ATCGEN are limited to maneuvers that are flyable by the aircraft. The main command input is the required acceleration $\dot{V}_{SS}$, and the output is the acceleration $\dot{V}_{SC}$. TRACOM effectively limits command inputs to changes the aircraft can follow by generating error signals that counteract large input changes as well as direct limiting on the command variables.

The actual computations performed by TRACOM can best be shown by reference to the series of block diagrams that follows. The TRACOM control law portion is shown in figure 14. As can be seen in figure 14, the ATCGEN inputs

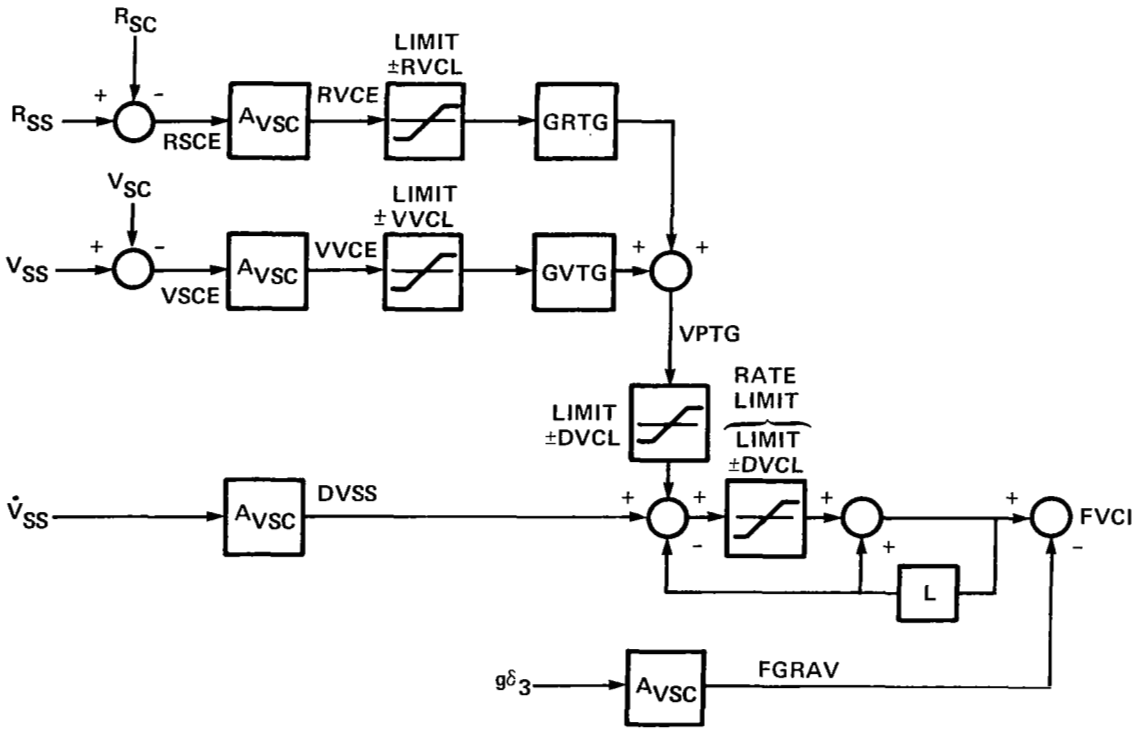
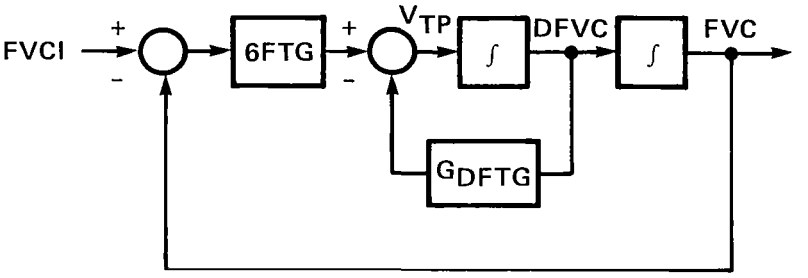


Figure 14.- TRACOM control law.

come in at the left side of the drawing. The position and velocity commands are summed with the TRACOM equivalent, R_{SC} and V_{SC}, to form error signals. These error signals, along with the acceleration input, are transformed to the commanded velocity frame with the direction cosine matrix derived in equation (A13) and then passed through the various limiters and gains as shown. The acceleration signal is summed with the limited error signals and passed through a rate limiter to produce an augmented acceleration command for the aircraft control. The acceleration due to gravity is then subtracted, leaving a commanded specific force, F_{VCI}, that is to be generated by the aerodynamic forces from the vehicle and the engine thrust. This specific force command is given in the commanded velocity frame and represents an open-loop command for the aircraft to follow. As can be seen from the diagram, if the ATCGEN command is smooth and if it is flyable by the aircraft, no position or velocity errors will develop and no rate limiting will be required. The result is that the output F_{VCI} is simply the acceleration commanded by ATCGEN. If, however, an error between the ATCGEN signals and the current TRACOM command does develop, the acceleration command will be modified to compensate for it. The rate at which this error can be reduced will be restricted because of the gains and limiters and will not require aircraft response beyond the capability of the vehicle. For the current tests with the Twin Otter, the limits were set for a rather soft system with the closure rate restricted to about 12 m/sec.

The TRACOM force servo computation is represented by a three-axis dynamical system having the servo structure shown in the sketch below. The operation



of the force servo is such that the rapidly changing input F_{VCI} is smoothed to make it consistent with the limitations of the force generation process of the aircraft. At the time of initiation of TAFCOS, the TRACOM states are loaded with the estimated aircraft states. Thereafter, TRACOM converges to the trajectory specified by ATCGEN. TRACOM outputs a flyable trajectory given by R_{SC}, V_{SC}, and $\dot{V}_{SC}$ and corresponding specific force F_{VCI}. For smooth aircraft operation, F_{VCI} contains a desired level of lead information relative to F_{VC}.

The specific force command will now be used to generate the required attitude command for the inner-loop operations. Angular velocity and angular acceleration will also be required to complete the command structure. These are not available from the trimmap operations and must be obtained from some other source. A good approximation of the needed quantities can be obtained

from the motion of the velocity coordinate system that has been used in the TRACOM computations. For the angular rate computations,

$$\omega_s = \begin{vmatrix} \frac{FVC(2)}{EV_2C} \tan \gamma_v \\ \frac{-FVC(3) + g \cos \gamma_v}{EVC} \\ \frac{FVC(2)}{EVC} \end{vmatrix} \tag{A14}$$

The angular acceleration may be obtained from the following:

$$\dot{\omega}_s = \frac{1}{EV_c} \left\{ [EVC(\omega_s)] - \dot{EVC}(\omega_s) \right\} \tag{A15}$$

Substituting as required and regrouping gives

$$\dot{\omega}_s = \begin{bmatrix} 0 \\ \frac{-\dot{FVC}(3)}{EVC} \\ \frac{\dot{FVC}(2)}{EVC} \end{bmatrix} + \begin{bmatrix} \left(\frac{-\dot{FVC}(2)}{EVC} \tan \gamma_v - \frac{\omega_s(2)\omega_s(3)}{\cos \gamma_v} \right. \\ \left. - \frac{g}{EVC} \sin \gamma_s \omega(2) \right) - \frac{\dot{EVC}}{EVC} \omega_s \\ 0 \end{bmatrix} \tag{A16}$$

TRAPCO

The next block in the trajectory loop computations is TRAPCO, the trajectory regulator or trajectory perturbation controller (see fig. 13). It is in this computational block that trajectory feedback information is input to the TAFCOS control logic. The section is self-explanatory from the block diagram of the computations, figure 15, and is discussed here only briefly. Basically, the position and velocity errors as the difference between the values commanded by the trajectory command generator (again those predicted from the previous cycle) and the same quantities as supplied by the navigation routine or air data measurements, are constructed. These errors are transformed into the velocity frame through the direction cosine matrix (eq. (A13)), limited and summed as shown on the block diagram. The result is the signal UDVTP shown on the diagram. This signal is summed with the acceleration command from TRACOM, F_{VCI}, and then summed with a signal proportional to the integral of acceleration error.

The integrator operates on the total commanded acceleration, F_{VI}, and the measured equivalent from the aircraft sensors. Long-term disturbances or trimmap errors will produce a build-up in the integrator which thereby unloads

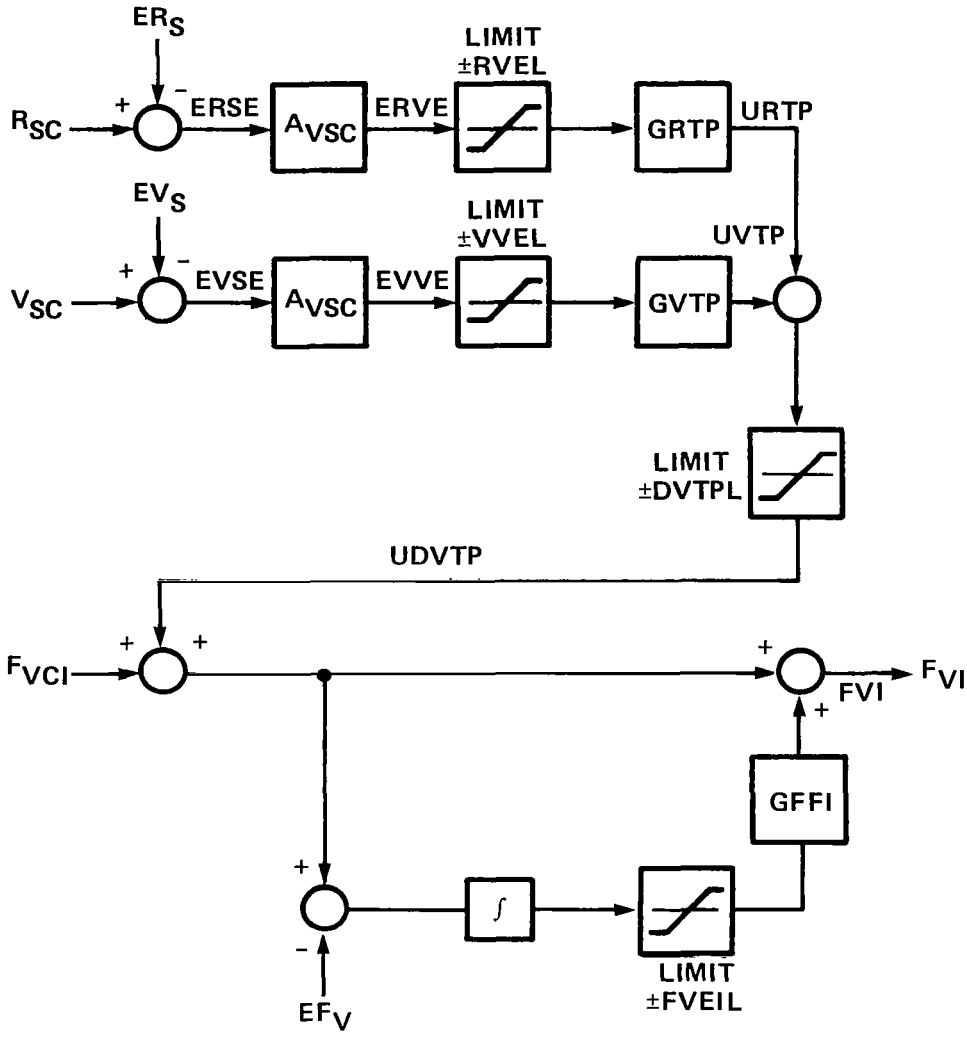


Figure 15.- TRAPCO block diagram.

the perturbation controller feedbacks. The new value of F_{VI} is the total commanded acceleration to fly the commanded flightpath; it is proportional to the total force required to be generated by the aircraft in order to follow the path command and is corrected for external disturbances or other errors due to internal modeling errors. The frequency response for the aircraft characteristics are also set by the perturbation controller by choice or the feedback gains.

Because all the variables required are now available, it is appropriate to discuss the path length reset for ATCGEN. The need for the reset comes about because the aircraft is flying an airspeed control flight trajectory. The computations in ATCGEN are all performed in the ground or inertial coordinate frame so that essentially a groundspeed control is being commanded.

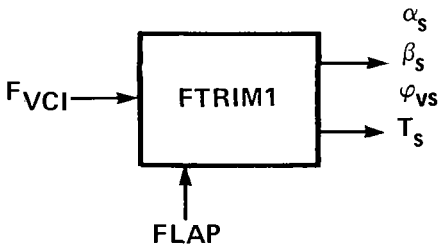
However, the velocity error behavior in TRAPCO is a function of airspeed. The result is an inconsistency between velocity and position for the components along the flightpath. To take care of this error build-up, the value of S_{ATC} in the ATCGEN routine must be reset as a function of the long track position error. To perform the reset operations, two equations are required; they are shown below:

$$\left.\begin{aligned} R_{SC} &= R_{SC} + A^{t}_{VSC}[K_{reset}(ERVE)] \\ S_{ATC} &= S_{ATC} - K_{reset}(RVCE) \end{aligned}\right\} \qquad (A17)$$

An error build-up in the position along the flightpath will be reflected in the TRACOM variable R_{SC} as well as S_{ATC} because one is a function of the other. The assumption in the equations is that a build-up of long track error is due to ground winds and to the required airspeed control. The position along the flightpath should be set at present vehicle location and then all ATC commands based on that position. The effect of the equations given above is to reduce the error by a small percentage each cycle to gradually move the commanded position from ATC into alignment with the vehicle position. Short-term disburbances will have a minimal effect but there will be a compensation for long-term effects, such as a steady ground wind.

TRIMMAP

The trimmap is the computational portion of TAFCOS; it contains the force modeling of the aircraft. In the trajectory loop, only those portions concerned with the trajectory variables are taken into consideration, and the task for the trimmap operation is to take the commanded specific force F_{VI} and solve for the aircraft attitude and throttle setting that will produce that acceleration for the particular flight condition and aircraft configuration that exists at the time. In the flight program, these computations are performed in the sections marked FTRIM1 (see sketch below).



For the modeling of the aircraft as used in the trimmap, a set of analytical equations was devised which were taken from a set of tabular data used in a piloted simulation of the Twin Otter aircraft. The trimmap routine could use the data in almost any manner. A table look-up routine could deal directly with the data arrays, but in the interest of simplicity, the data were reduced to the analytical format. In addition, only the major aerodynamic terms were included in the model. This simplification reduced the number of computations, and the attitude results could be solved directly without the need for iterative

solution routines. Clearly, these simplifications might not be valid for an aircraft more complex than the Twin Otter, and the trimmap operations might take on a more complex form.

The output, F_{VI}, from the perturbation controller is the drive signal for the trimmap routine. F_{VS} is the specific force required to follow the commanded path and compensates for disturbances or other errors. The corresponding force coefficient is given by the following equation:

$$C_{VC}(I) = F_{VI}(I) \frac{EMASS}{EQS} \tag{A18}$$

where EMASS is the aircraft mass and EQS is the dynamic pressure times wing area. The coefficients can be resolved into a pair of coefficients with one along the velocity vector and the other perpendicular to that vector. An associated "roll" angle, the roll angle about the velocity vector, must also be specified for the component perpendicular to the velocity vector. The roll angle will be required later.

$$\left.\begin{aligned} C_{DC} &= CVC(1) \\ C_{LC} &= [CVC(2)^2 + CVC(3)^2]^{1/2} \\ \phi_{VS} &= \tan^{-1} \frac{CVC(2)}{CVC(3)} \end{aligned}\right\} \tag{A19}$$

As is fairly clear from the nomenclature used, these coefficients can be considered as a thrust-drag coefficient and as a lift coefficient. In an approximate sense, the C_{DC} coefficient is proportional to the engine throttle setting, and the C_{LC} coefficient is proportional to the angle of attack to generate the lift. The quantity ϕ_{VS} is approximately the aircraft roll angle.

From the tables of data for the Twin Otter aircraft, the following equations can be written as an approximation of the aircraft characteristics:

$$\left.\begin{aligned} C_{DC} &= C_D(0) + \frac{\partial C_D}{\partial C_T} C_{T_s} + \frac{\partial C_D}{\partial C_L{}^2} C_{LC}^2 \\ C_{LC} &= C_L(0) + \frac{\partial C_L}{\partial \alpha} \alpha_S + \frac{\partial C_L}{\partial C_T} C_{T_s} \end{aligned}\right\} \tag{A20}$$

The partials and constant coefficients in the above equations are of the form shown below:

$$
\left.
\begin{aligned}
C_D(0) &= 0.037 + 0.170\ \delta F \\
\frac{\partial C_D}{\partial C_T} &= -0.910 + 0.575\ \delta F \\
\frac{\partial C_D}{\partial C_L{}^2} &= 0.044 - 0.030\ \delta F \\
C_L(0) &= 0.500 + (4.75 - 3.187\ \delta F)\delta F \\
\frac{\partial C_L}{\partial \alpha} &= 0.091 + 0.051\ C_{T_s} \\
\frac{\partial C_L}{\partial C_T} &= 0.560\ C_L(0)
\end{aligned}
\right\} \qquad \text{(A21)}
$$

where δF is the flap setting. Equations (A20) can now be used to solve the commanded thrust coefficient and the commanded angle of attack from the following set of equations:

$$
\begin{aligned}
C_{T_s} &= \left\{ - \frac{\partial C_D}{\partial C_L{}^2} + C_{LC}^2 + [-C_D(0) + C_{DC}] \right\} \Big/ \left(\frac{\partial C_D}{\partial C_T} \right) \\
\alpha_s &= \left\{ - \frac{\partial C_L}{\partial C_T} + C_{T_s} + [-C_L(0) + C_{LC}] \right\} \Big/ \left(\frac{\partial C_L}{\partial \alpha} \right)
\end{aligned}
\qquad \text{(A22)}
$$

The s subscript is used on α_s, C_{T_s}, and ϕ_{v_s} as these quantities are the equivalent of ATC commands to the attitude control system.

With a knowledge of engine characteristics, the thrust coefficient can now be converted to a throttle setting. The engine thrust can be modeled by the following equation in which the thrust is defined for both engines and T_s is the throttle handle position:

$$
\text{Thrust} = T(0) + \frac{1}{473} \left. \frac{\partial T}{\partial t} \right|_{v=0} (v_0 - V)[T_s - T_s(0)] \qquad \text{(A23)}
$$

where

$$T(0) = 816$$

$$v_0 = 473$$

$$\frac{1}{v_0}\frac{\partial T}{\partial t} = 5.074$$

$$T_s(0) = 4.0$$

a minimum position of the throttle handle. For use by the trimmap, the equation above is solved for the throttle handle position as shown below.

$$T_s = \left\{[C_{T_c}(EQS) - 816]0.1970\right\} \Big/ [(473 - V_{CAL}) + 4] \qquad (A24)$$

where V_{CAL} is the measured calibrated airspeed. Again, EQS is the dynamic pressure times wing area.

The attitude command is, of course, incomplete if only the angle of attack command is defined. The necessary quantities to completely define the commanded attitude are given by the equation below where A_{cs} is the direction cosine matrix between the ground reference frame and the commanded body frame.

$$A_{cs} = E_2(\alpha_s)E_3{}^T(\beta_s)E_1(\phi_{vs})E_2(\gamma_v)E_3(\psi_v) \qquad (A25)$$

It is assumed that the maneuvers required of the aircraft will always be made without side slip (zero β_s). With the previously defined values of ϕ_{vs}, γ_v, and ψ_v, the attitude command is completely defined.

Attitude Control System

The computations performed in the attitude loop are generally similar to those made in the trajectory loop, except, of course, that the variables are there concerned with rotation control. In addition, some operations are performed on the throttle commands. The output of the trajectory loop, the attitude variables α_s, β_s, and ϕ_{vs}, the angular rate and acceleration, ω_s and $\dot{\omega}_s$, and the throttle command T_s, can be considered as commands similar to those generated by ATCGEN for the trajectory loop. The computational objective of the attitude control system is to take the commanded angular acceleration, which may be considered as a commanded moment, and generate the control settings to produce that moment and hence the commanded rotation. The computational sections in the attitude loop follow the format of the trajectory commands with an attitude command generator in sections labeled ROTCOM and ROTINT; a perturbation controller, ROTPCO and ROTCON; a trimmap, MOTRIM; and the additional conversion section called SERVOS. A block diagram of the attitude loop operations is shown in figure 16.

ROTCOM

The attitude command generator performs a function similar to that of the trajectory command generator where the commands from the trajectory loop are smoothed as required and converted to a set of rotation commands that are flyable by the aircraft. The need for the smoothing comes about from the

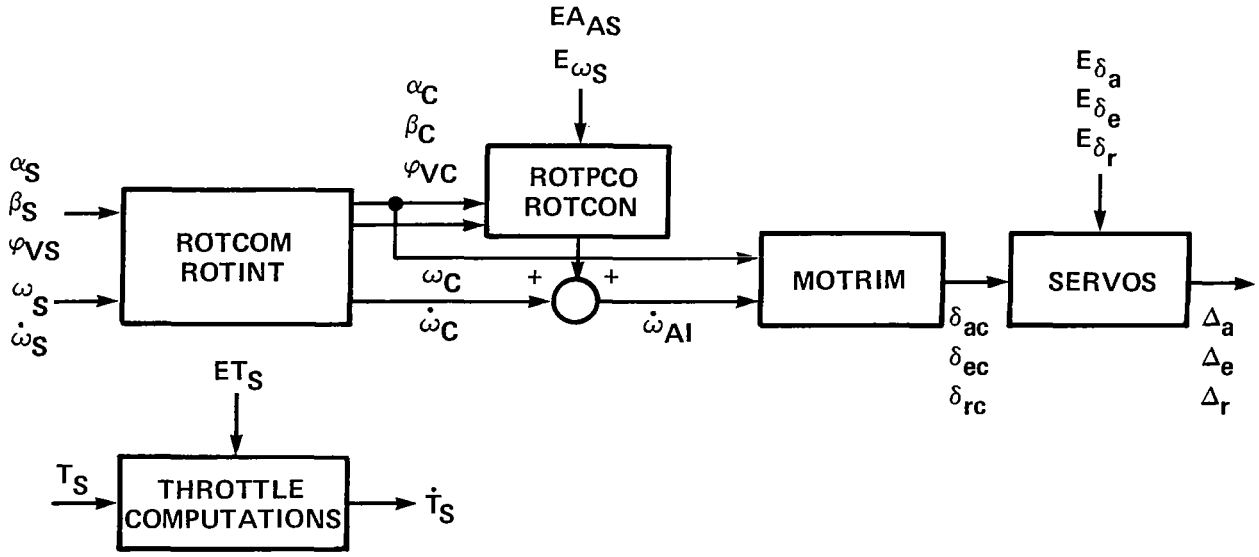
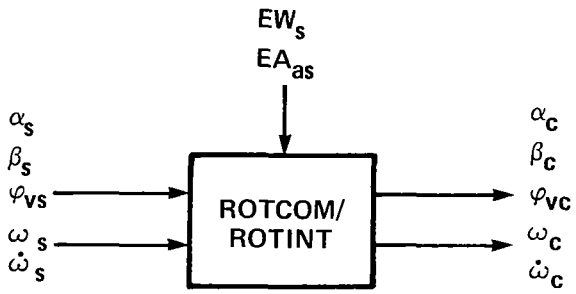


Figure 16.- Block diagram of attitude loop interconnections.

difference in cycle time between the trajectory and attitude loops. As mechanized on the Twin Otter aircraft, the attitude loop operates at five times the rate of the trajectory loop. The result is that the inputs to the attitude loop are coarse steps and some smoothing is required. The operation of the ROTCOM computations is shown on the block diagram below.



ROTCOM can be conceptualized as shown in figure 17.

As shown in the figure, the operations are essentially the same as those carried out in the TRACOM section of the trajectory command generator. In general, the operation of the ROTCOM computations provides for an augmented angular acceleration command from angular rate and attitude errors (see fig. 18 for the control law logic). The feedback terms α_C, β_C, ϕ_{VC}, and ω_C are the values predicted from the previous computation cycle.

The remaining portion of the command generator consists of the computations required for the predicted values of attitude and rate commands for the next cycle.

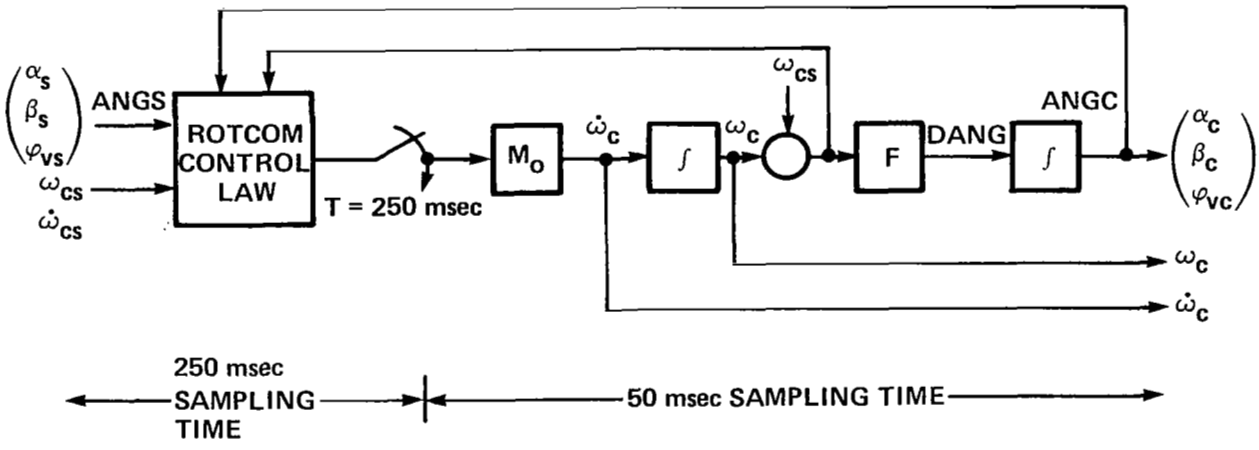


Figure 17.- Structure of ROTCOM.

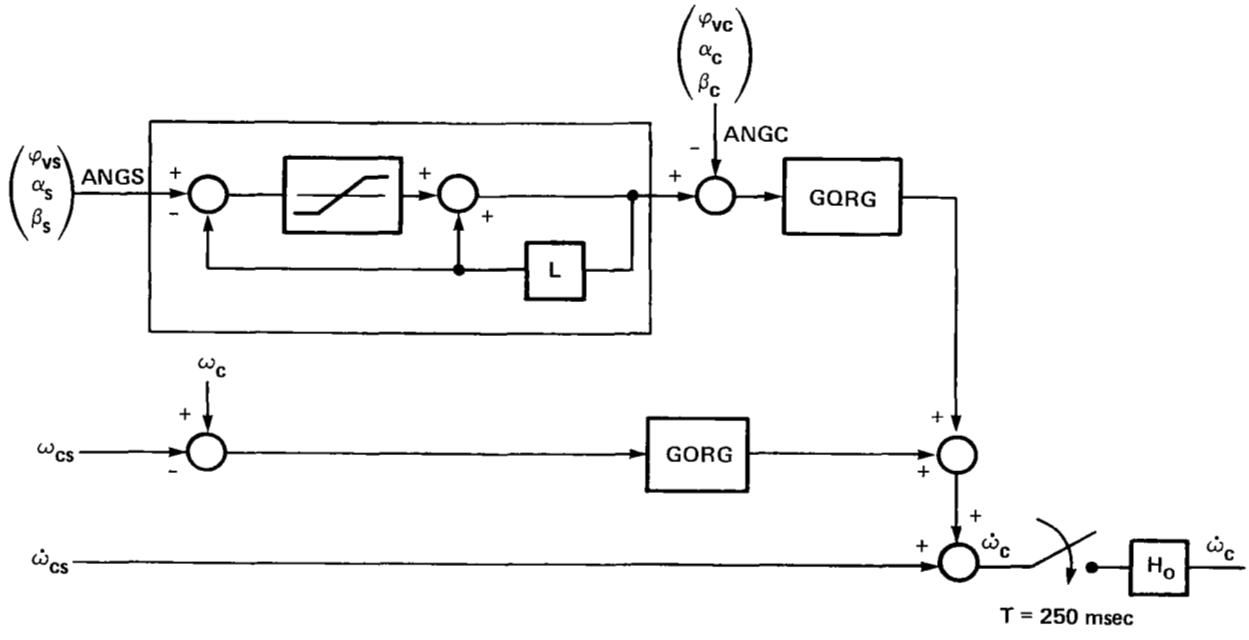
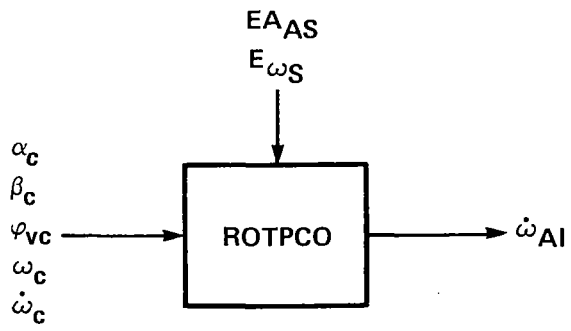


Figure 18.- ROTCOM control law.

ROTPCO

The attitude perturbation controller is the location in TAFCOS where attitude feedback is introduced. The input/output block diagram is shown below. The basic structure of the perturbation controller is quite similar to



that used in the trajectory loop and provides for an augmented angular acceleration command signal that is a function of attitude error, angular rate error, and the integral of angular acceleration error. The block diagram of the computations is shown in figure 19.

The structure of the attitude perturbation controller differs in two main respects from the equivalent trajectory computations. The first is that the position error must be handled differently because the command is in the form of α_c, β_c, and ϕ_{vc} where the feedback is a direction cosine matrix of vehicle attitude. Also, the feedback terms are obtained at the slow cycle rate and require smoothing to be used as the feedback term. The other major difference is in the operation of the integral of angular acceleration. Measured angular acceleration is not available from instrumentation on board the aircraft, so that quantity must be derived from angular rate information for the integration.

The attitude feedback computations can be seen in the upper right corner of the block diagram on figure 19. The feedback quantity is the direction cosine matrix EA_{as}, which is obtained from instrumentation on board the aircraft. Basically, what is wanted is an estimate of the angle of attack, side slip, and roll angle, rather than the full attitude. The direction cosine matrix defining the velocity coordinate system was derived in the section marked COMVA and can be used to extract the desired aerodynamic variables. A direction cosine matrix can be obtained by the following, which is a function of $E\alpha$, $E\beta$, and $E\phi_v$:

$$A_{av} = EA_{as}E_3{}^T(\psi_v)E_2{}^T(\gamma_v) = EA_{as}A_{vsc}^T \tag{A26}$$

Then, specific terms within the matrix can be used to extract the desired variables by use of the following relations:

$$\left.\begin{aligned} E\phi_v &= \delta_2 A_{av}\delta_3 \\ E\alpha &= S_3{}^T A_{av}\delta_1 \\ E\beta &= -\delta_2{}^T A_{av}\delta_1 \end{aligned}\right\} \tag{A27}$$

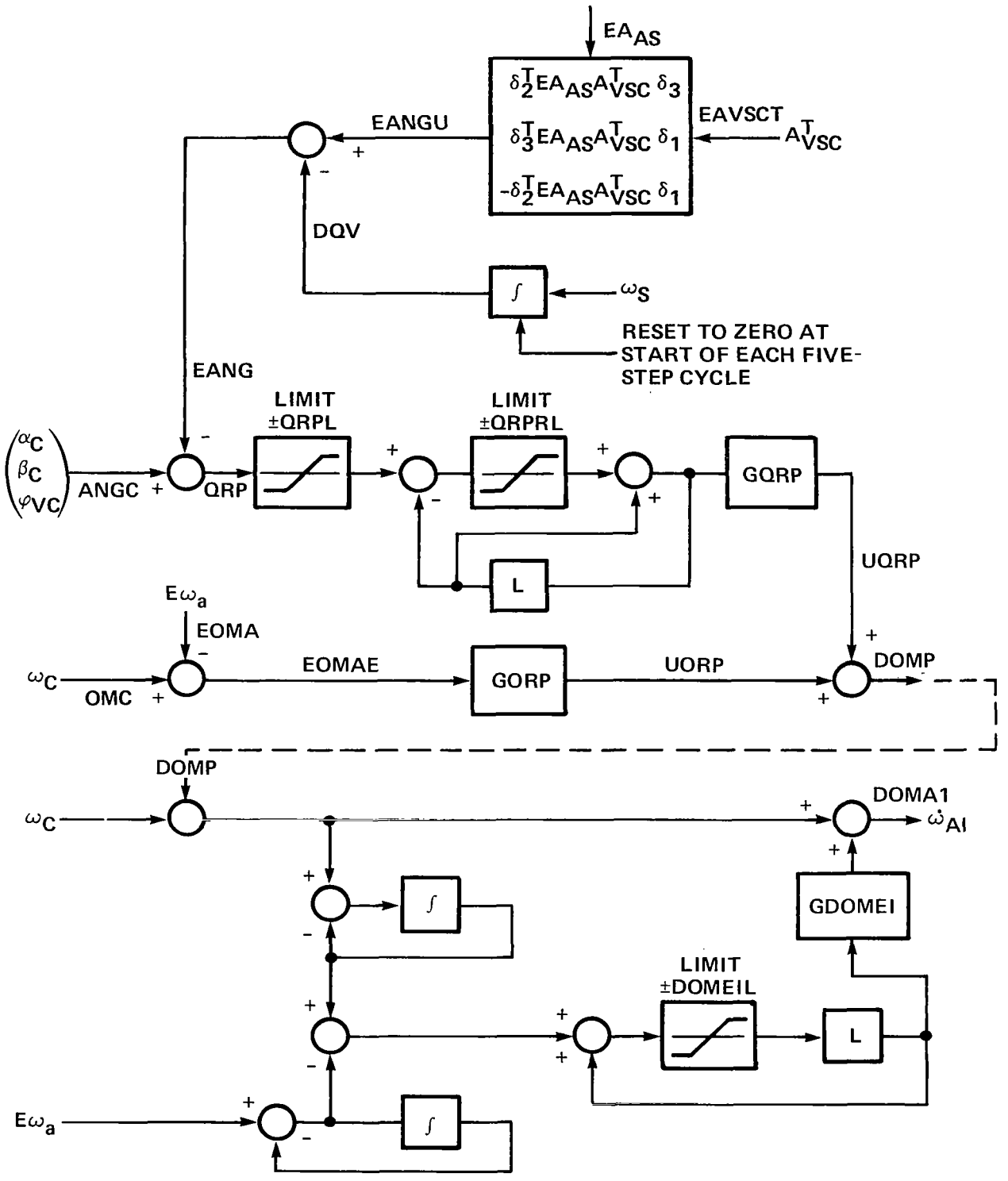
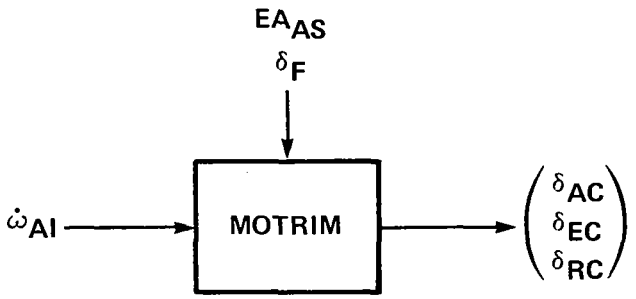


Figure 19.- ROTPCO computations.

The angular acceleration commands are converted to control surfaces commands through an attitude trimmap called MOTRIM. Like the force trimmap, MOTRIM contains an analytical model of the aircraft; the model was constructed to solve for the aileron, elevator, and rudder surface positions, given a commanded moment for them to generate. The input/output variables for this section of the program can be seen from the block diagram below.



The equations solved by MOTRIM can be seen from the following set of equations. It has been assumed that the moment equations for the Twin Otter aircraft can be described in the following form:

$$C_{mc} = (A_1)\begin{pmatrix} 0 \\ E\alpha \\ E\beta \end{pmatrix} + (A_2)E\omega_a + (A_3)\begin{pmatrix} \delta_{AC} \\ \delta_{EC} \\ \delta_{RC} \end{pmatrix} \tag{A28}$$

where the quantities A_1, A_2, and A_3 contain dynamic information on the aircraft. In equation (A28), a number of aerodynamic variables and the gyroscopic terms on the left side of the equation have been assumed negligible. Solving these equations for the control surface commands gives the following set of equations to be mechanized in MOTRIM:

$$\begin{pmatrix} \delta_{AC} \\ \delta_{EC} \\ \delta_{RC} \end{pmatrix} = \left[(A_3^{-1})\ C_{mc} - (A_1)\begin{pmatrix} 0 \\ E\alpha \\ E\beta \end{pmatrix} - (A_2)E\omega_a \right] \tag{A29}$$

The several matrices in the equation are a function of the aerodynamic properties of the Twin Otter aircraft and are given by the set of numbers below.

$$A_1 = \begin{pmatrix} 0 & 0 & -0.100 \\ 0 & -1.891 & 0 \\ 0 & 0 & 0.131 \end{pmatrix} \tag{A30}$$
Cont'd

$$A_2 = \begin{pmatrix} -0.55 & 0 & a_{13} \\ 0 & -24.0 & 0 \\ 0 & 0 & -0.20 \end{pmatrix} \qquad \text{(A30) Conc'd}$$

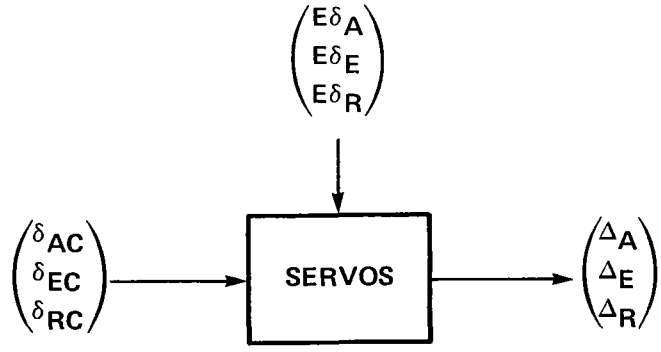
where $a_{13} = 0.15 + 0.579\ E\alpha + 0.874\ \delta F.$

$$A_3^{-1} = \begin{pmatrix} 5.01 & 0 & 1.267 \\ 0 & -0.5714 & 0 \\ 0 & 0 & -6.667 \end{pmatrix}$$

With the approximations made in the moment equations for the Twin Otter aircraft, the solution of the control surface settings given a moment command are quite straightforward with no iterations or other computational problems. For another aircraft, or for more precise control of the Twin Otter, a table look-up type operation may be needed, or possibly some iterative routine would be required.

SERVOS

The final section of the attitude control portion of the attitude loop is the conversion of the control surface position commands to the delta form required by STOLAND. Clearly, such a section is vehicle-specific; it might not be required for another control setup or might possibly be in a very different form. For the STOLAND-Twin Otter setup, the most straightforward way to supply the commands to the aircraft was to utilize the existing control channels and the STOLAND format. The input/output requirements for this computational block are shown below.



The necessary computations to carry out the conversion of the control surface commands are shown by the block diagram in figure 20. The computations call for a rate limit on the control surface commands, so that they do not exceed the capability of the aircraft servos, and then use a form of estimator for the measured control surface position in order to generate the delta commands. An estimator was used rather than simply taking the difference between

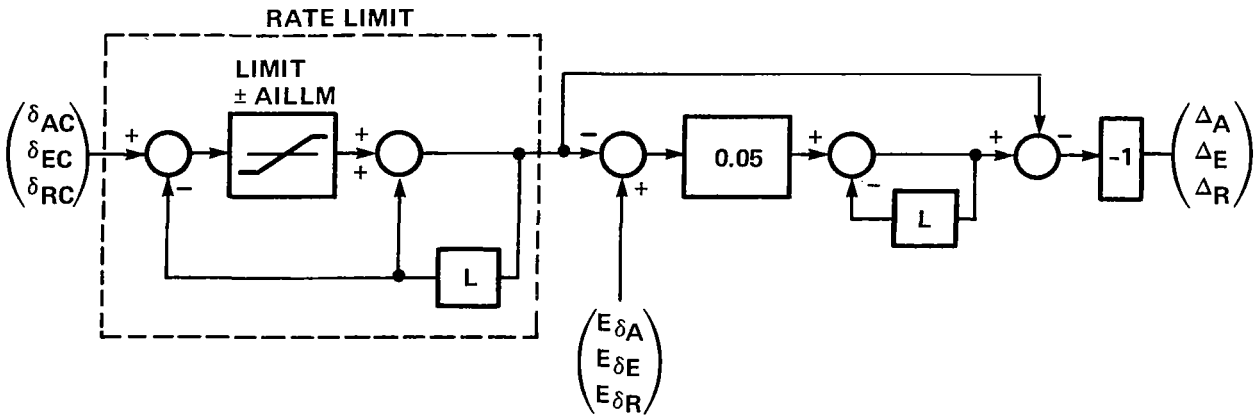


Figure 20.- Servo computations.

the commanded and measured positions because of a data sampling problem within the computer. The outputs from the servos section are the commands to the vehicle servos and directly drive the control surfaces.

Throttle Computations

The attitude loop computations performed thus far have been concerned only with the aerodynamic control surfaces, but throttle commands are also needed for a construction of the total force command. Because the throttle command is an output of the trajectory loop, the slow timing of this loop will cause the throttle handle position command to be coarse. Smoothing is required just as was done with the attitude commands; in addition, the STOLAND system operates with a throttle rate command so that a conversion to that form was required. The throttle command computations are somewhat simpler than those required for the attitude variables, and the complete operation is handled as one computational sequence. However, one can view the operations as having all the operations performed on the other variables with a throttle command generator, perturbation controller, trimmap, and servos section. A block diagram for the sequence of operations is shown in figure 21. The throttle handle command from the trajectory loop is shown by the quantity T_s on the left side of the diagram. The processing required to convert this signal to T_c can be considered as the command generator section where T_c is the smoothed throttle handle command for the aircraft. The remaining computations bring the throttle handle feedback information from the aircraft and generate the throttle rate command information required by STOLAND. The final throttle rate signal is then passed through a limiting routine that restricts the signal based on engine torque pressure limits imposed as a safety feature for the engines. With computations of the throttle rate command, all the necessary control inputs are available for the aircraft to fly the specified ATC trajectory. The only control items not handled through TAFCOS are the flap setting and the propeller rpm. Neither of these items is servo-controlled on the Twin Otter aircraft; however, if they were, control outputs could be readily provided by TAFCOS.

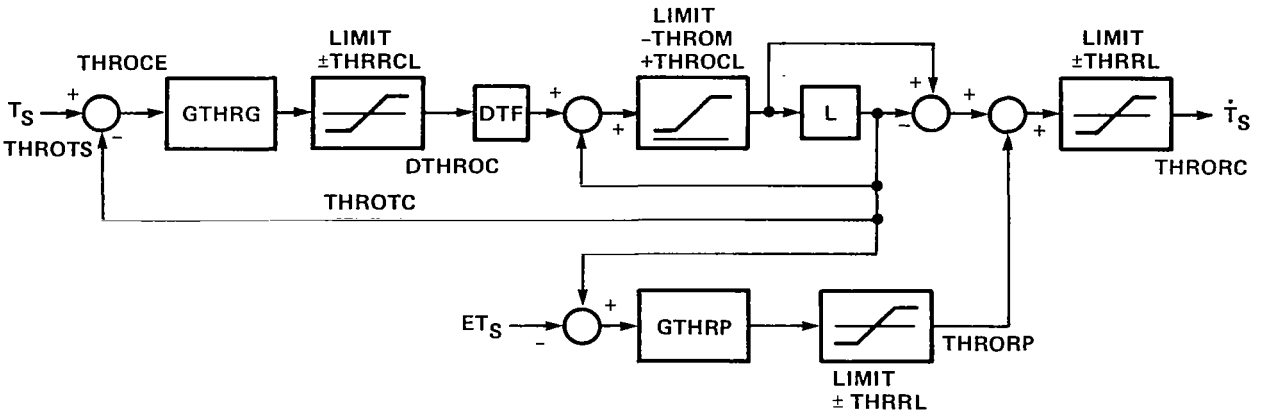


Figure 21.- Throttle computations.

APPENDIX B

SIMULATION/FLIGHT EQUIPMENT AND PROGRAMMING

STOLAND System

The Ames STOLAND system is a set of avionics and software that was constructed by Sperry Flight Systems. STOLAND is a flexible integrated digital avionics system for performing navigation, guidance, control, and display experiments on a STOL research aircraft, in the present work a DHC-6 Twin Otter aircraft. The STOLAND system includes as a main computational device, a Sperry 1819A digital flight computer with an auxiliary memory (total of ~32 k). Through a data adaptor, the 1819A communicates with the outside world, receiving sensor signals from a wide range of devices and outputting information for servo drives of the aircraft control surfaces, pilot displays, and a data collection system for the researcher. A block diagram of the system is shown in figure 22.

On the output side of the STOLAND system, drive signals are provided for servo control of the ailerons, elevator, and rudder through a parallel servo system. Power control is also provided through an autothrottle system. Of the major control items for the aircraft, only propeller rpm and flap setting are manually controlled by the pilot (STOLAND has flap control outputs but the aircraft is not mechanized to use them). Signals are also supplied to a number of displays for pilot monitoring of system performance as well as manual control. The displays include a CRT EADI for flight director commands, a CRT moving-map display called an MFD for an X-Y situation display, a conventional HSI, and other status information shown by a number of switches and lights. A photograph of the simulator cockpit is shown in figure 23. Only small differences exist between the simulator cockpit and the Twin Otter instrumentation.

The inputs to the STOLAND system come from a variety of sources. Navigation information can be received from either VOR/DME or TACAN stations for enroute flight and from an ILS/LOC or MODILS landing system for final approach guidance. Measurements are provided of many aircraft states, such as center of gravity, acceleration, aircraft attitude, various air data values, and control surface positions. The pilot can communicate with the STOLAND system with either a keyboard input, with which a limited number of gain changes, etc. can be entered, or through a flight-mode select panel (MSP). The keyboard has a one-line display to show the input as typed in; that display can be used by the computer to display diagnostic and monitoring information for the pilot.

As delivered by Sperry, the STOLAND system includes a complete software package which performs the computations required for flight control of the aircraft. The control functions included are those of area navigation, various autopilot modes, including three-dimensional and four-dimensional operation, computations for an SAS system, and a variety of computations for display operation and data collection. The navigation computations are made using a complementary filter for aircraft position and velocity and wind estimation.

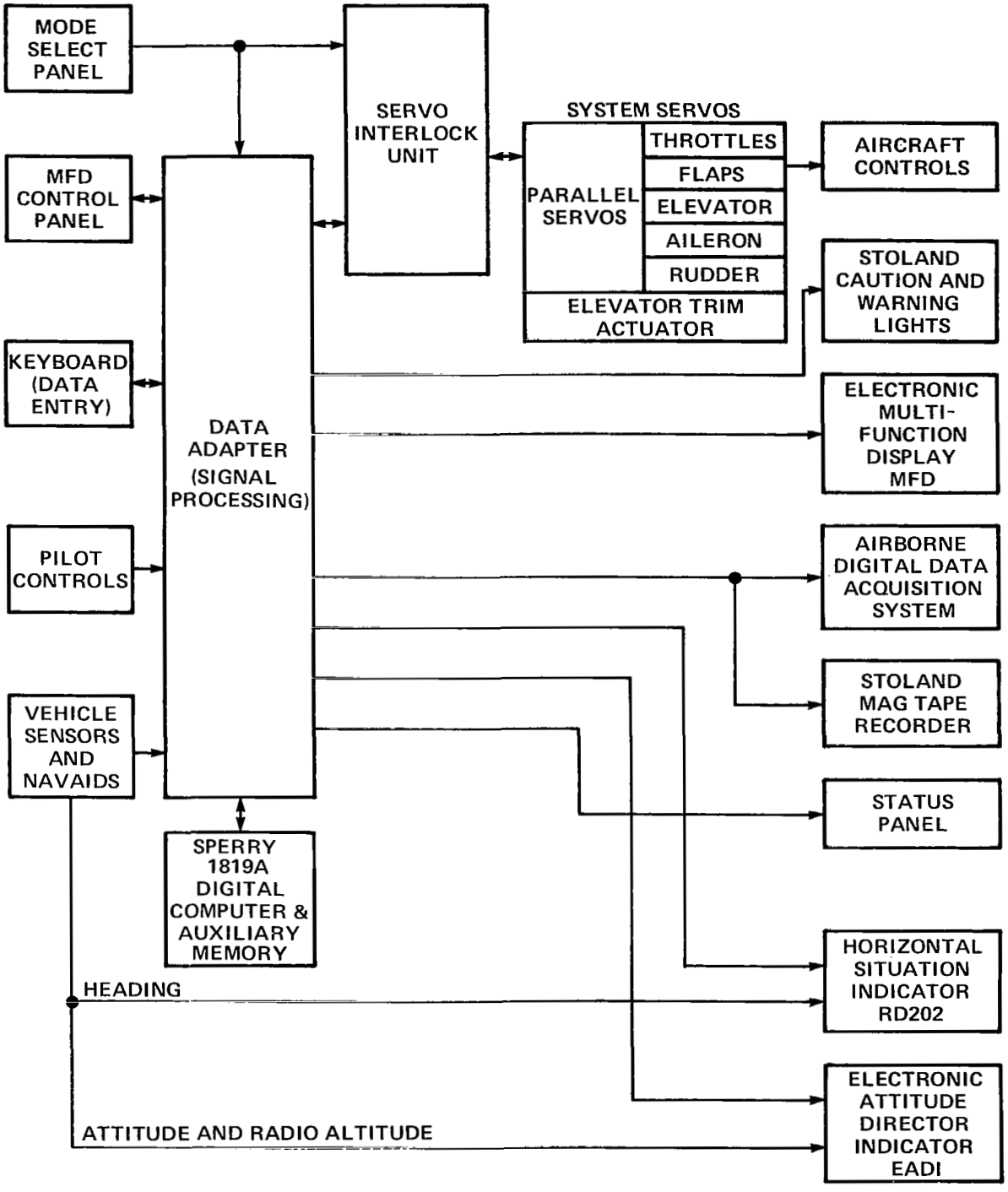


Figure 22.- Block diagram of STOLAND system.

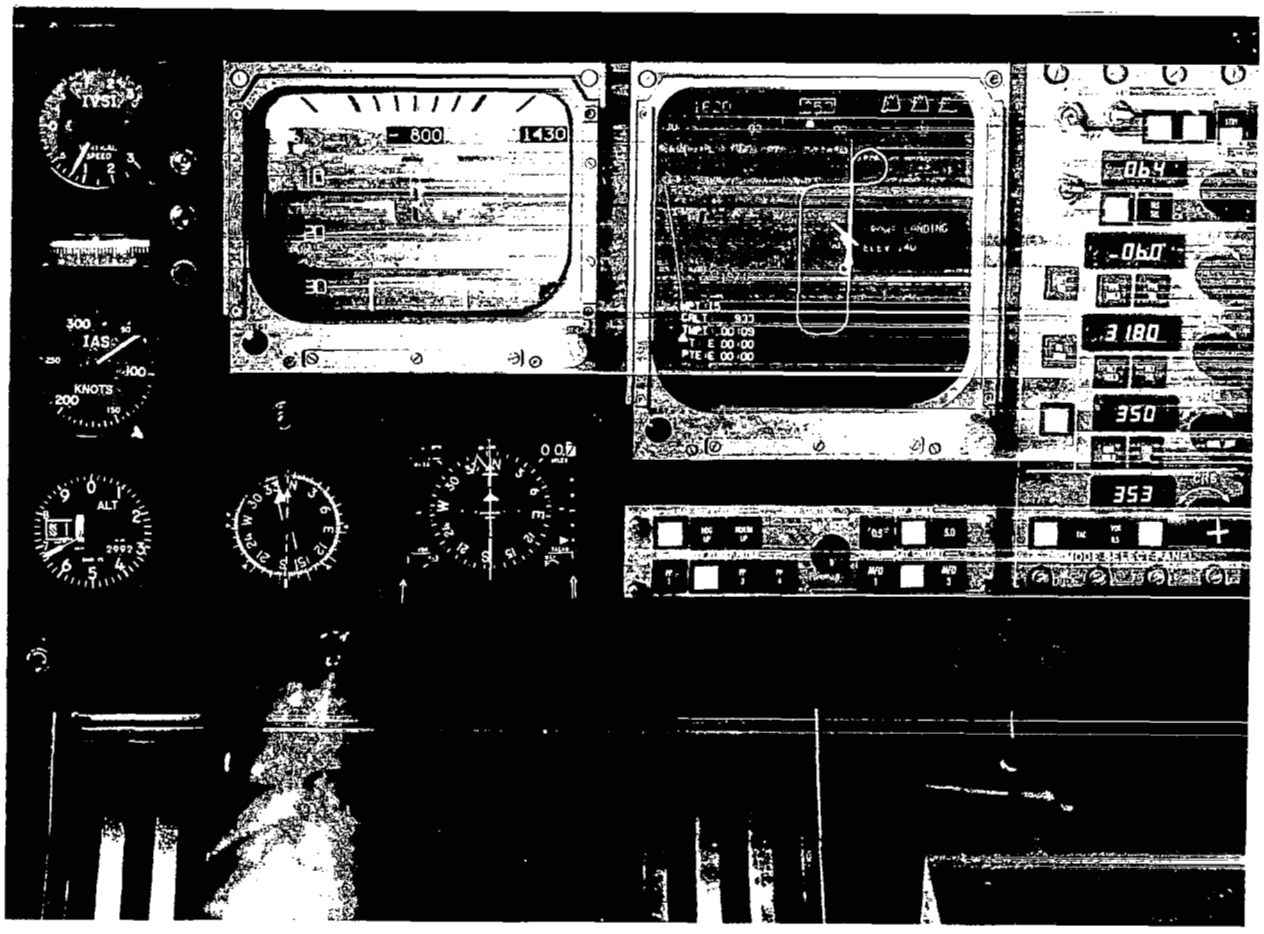


Figure 23.- Photograph of simulator cockpit.

For the RNAV functions, a set of stored way points is used to generate the track to be followed. The autopilot modes are, in general, reasonably conventional with airspeed hold, altitude hold, flightpath-angle hold, and heading hold, and with a full auto mode for the three-dimensional and four-dimensional control. A flow diagram for the computer executive logic is shown in figure 24. Details of the operation of the control logic are given in reference 3 and are not repeated here.

TAFCOS/STOLAND Interface

To flight-test the TAFCOS flight control system, it was decided to take the software package for TAFCOS and insert it into the STOLAND software package, retaining as much as possible of the STOLAND system. In particular, it was decided to use the navigation routine as provided by STOLAND for the estimates of aircraft position and wind velocity estimates and to use as much of the display outputs as was possible in exactly the mode used by STOLAND. The displays are not an integral part of TAFCOS, but it was found that using them

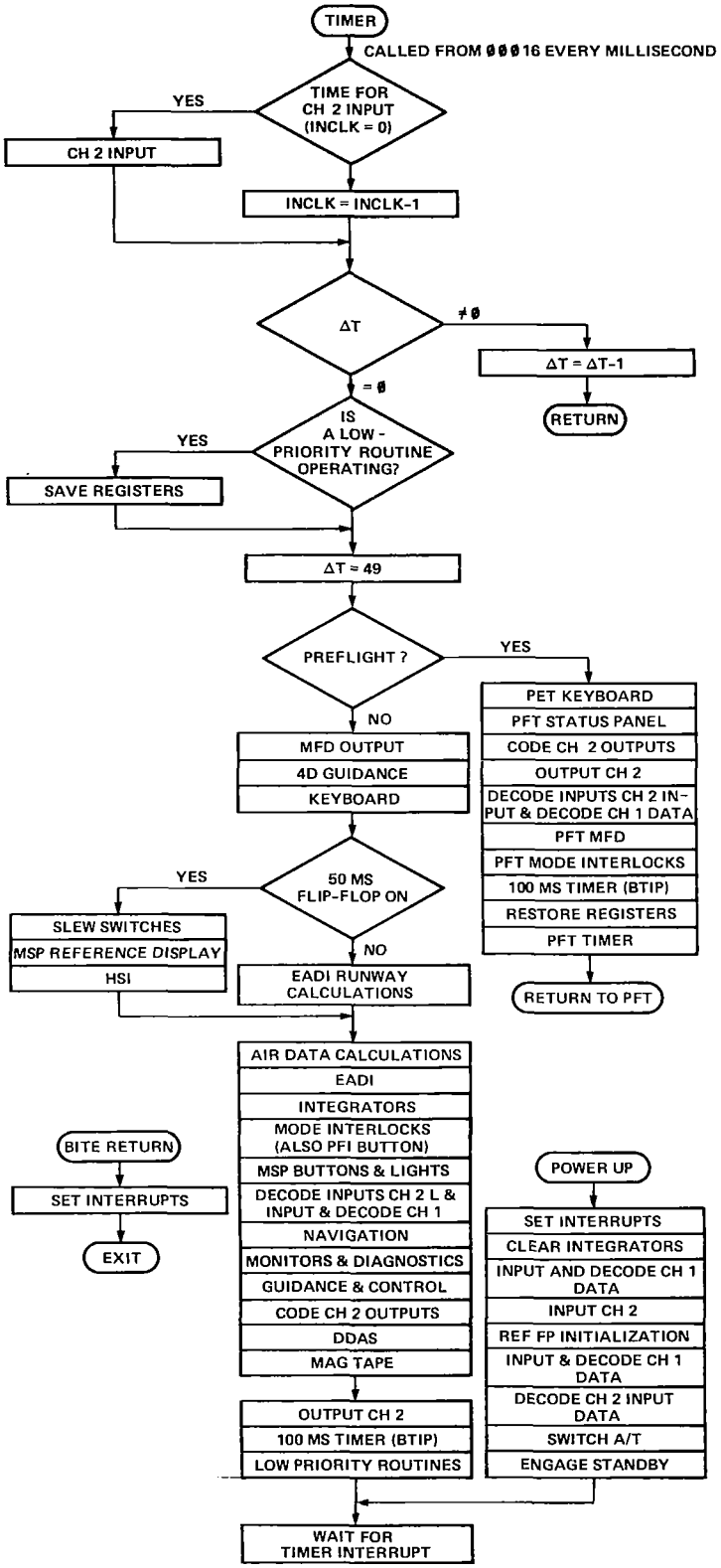


Figure 24.- Flow diagram of STOLAND executive.

in an essentially conventional manner provided an excellent means of giving status information to the pilot. The required input-output information for TAFCOS is given below.

Input

Program label	Definition	Source
ZN(1)	$\dot{X}$	
ZN(2)	X	
ZN(3)	$\dot{Y}$ Inertial	NAV
ZN(4)	Y	routine
ZN(5)	$\dot{Z}$	
ZN(6)	Z	
XDDRI	$\ddot{X}$	
YDDRI	$\ddot{Y}$ Inertial	Sensors
ZDDRA	$\ddot{Z}$	
VTAIRF	True air speed	Air data
VCAIRF	Calibrated airspeed	
Q	Dynamic pressure	
ZC15	X wind estimation	NAV
ZC16	Y wind estimation	routine
DELFLP	Flap setting	Sensor
THTDOT	• q	
PHIDOT	p	Sensor
PSIDOT	r	
SINTHT		
COSTHT		
SINPSI	Attitude	
COSPSI	measurements	
SINPHI	for direction	Sensor
COSPHI	cosine	
DCM13	construction	
DCM32		
DCM33		
THROT	Throttle position	
ELVPOS	Elevator position	
WHLPOS	Aileron position	
RUDPOS	Rudder position	

Output

Program label	Purpose
DELECS	Elevator command
DELACS	Aileron command
DELTDC	Throttle command
DELRCS	Rudder command
DELRFD	Display elevator servo error
DELPFD	Display aileron servo error
EPSY	Display lateral deviation
EPSZ	Display vertical deviation

Output - Concluded

Program label	Purpose
VCERR	Display speed error
IASDSP	Display commanded airspeed
FPADSP	Display commanded flightpath angle
ALTDSP	Display commanded altitude
PSIDSP	Display commanded heading
WPN	Display next way point
CRSDSP	Display commanded course
DELFCS	Flap command (not mechanized)

The integration of TAFCOS with the STOLAND software was accomplished as shown in figure 25. Essentially, TAFCOS was added as another mode of operation on top of all the STOLAND modes. A push button switch was selected on the MSP panel as a control device and, with the system operating in any of the fully automatic STOLAND modes (i.e., altitude hold, etc.) pushing the button caused the computational flow to go through the TAFCOS routine rather than the STOLAND routines. The same output channels used by the STOLAND system were also retained, although some modification in TAFCOS was required for operation. STOLAND is set up to receive delta commands for the aileron, elevator, and rudder; TAFCOS commands actual surface position. Also, the STOLAND system outputs throttle rate command as opposed to throttle position. Suitable additions were made to TAFCOS so that the outputs were of the STOLAND format.

TAFCOS Programming

TAFCOS programming is given in figure 26. The labels in each of the blocks are, in general, those identified in the theoretical discussion in appendix A.

The structure shown in the flow diagram is the same as that described in the main text of the report. The computations can be considered as divided into three main parts: the computation of the ATC trajectory commands, the outer or trajectory loop, and the attitude loop. ATC and the trajectory computations compose the slow loop and the attitude computations the fast loop. The cycle time of the 1819A is 50 msec and the complete attitude computation is done each cycle. About 20% of the ATC and trajectory computations are done each cycle, with a complete update on the commands each 0.25 sec. Starting from the top of the flow diagram, the first task is to get an update on the various state measurements needed for the TAFCOS computations from the navigation routine, sensors, etc. The routine called GETMES carries out this task. The controller then computes either the ATC commands or a portion of the trajectory loop depending on the setting of a flag called "cycle." The remainder of the computation is that of the attitude loop with the addition of a portion called SERVOS and DISPLAY. The SERVOS routine takes the TAFCOS output of control surface commands, transforms it into difference commands, and outputs it to STOLAND. The DISPLAY section outputs variables to drive the STOLAND display for pilot monitoring information. In addition to the above, a flag called IC is used to initialize quantities within TAFCOS for a smooth

transition from whatever mode the STOLAND was in at the time of TAFCOS turn-on. The IC sequence uses five cycles, as does the normal computation; the control commands are held fixed during this time. Reference 6 contains a complete listing of the TAFCOS program, and the various block calls can be seen in the small executive routine at the beginning of the program.

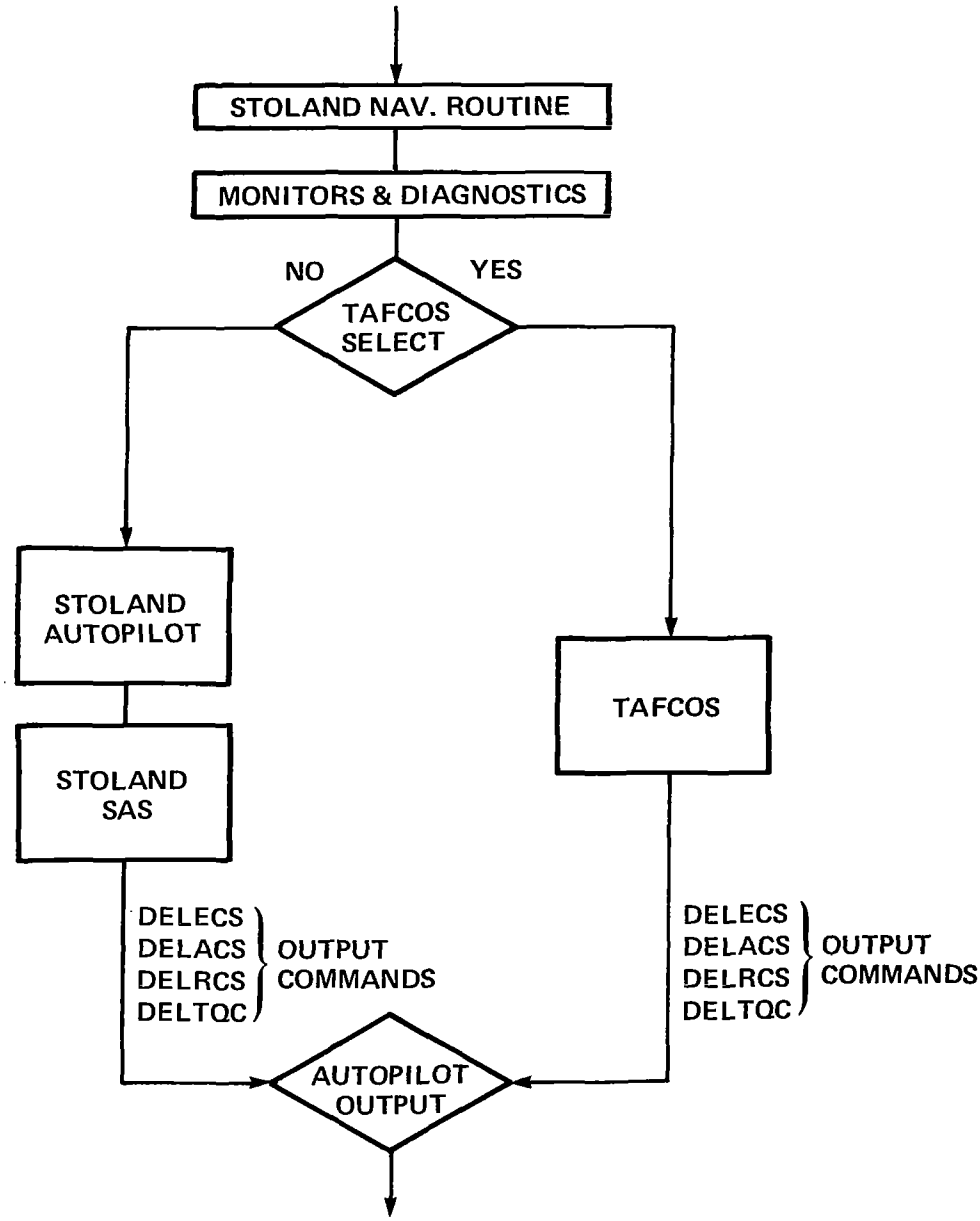


Figure 25.- TAFCOS/STOLAND interface.

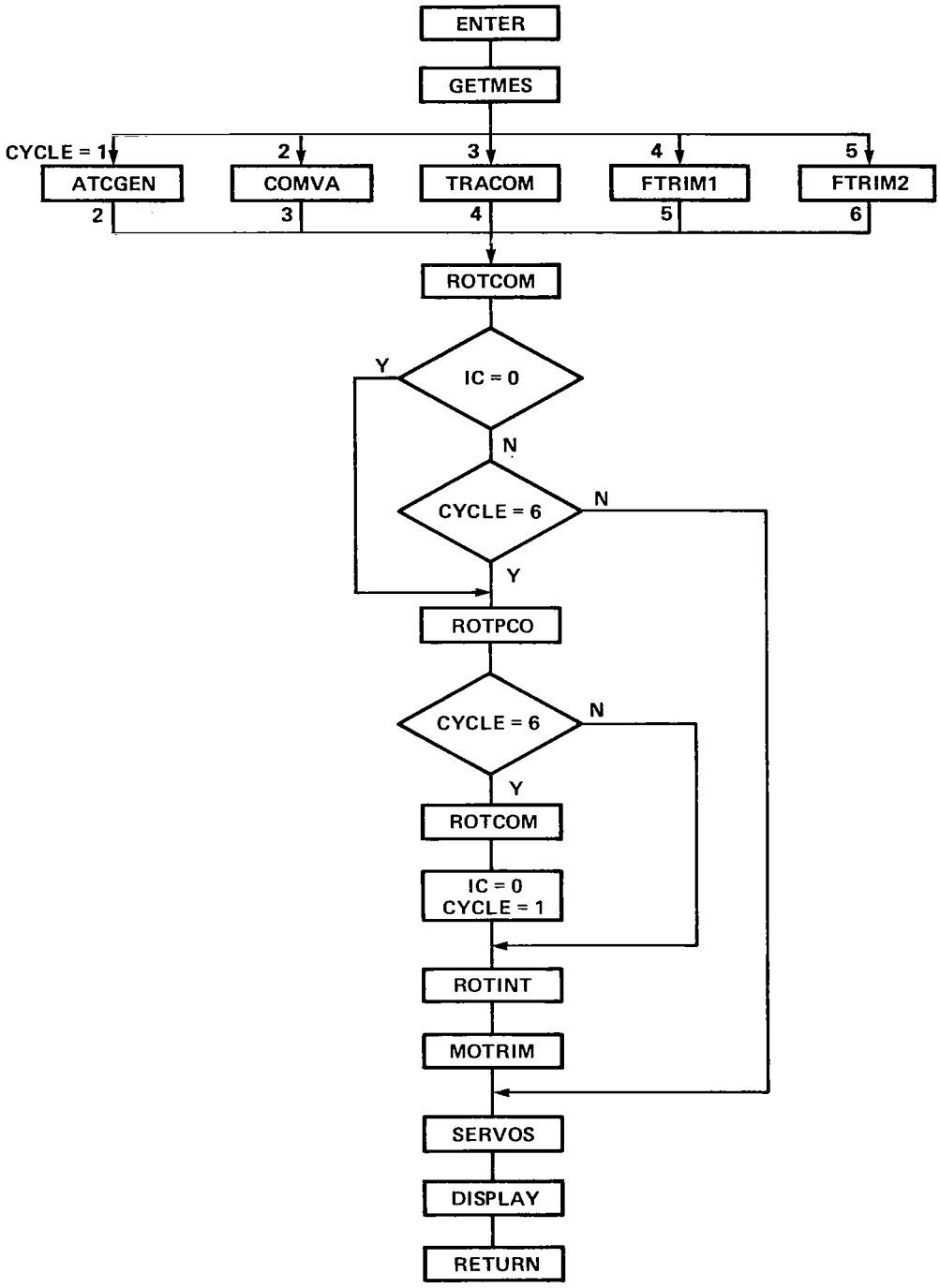


Figure 26.- Flow diagram for TAFCOS programming.

APPENDIX C

FLIGHT DATA

This appendix contains a full set of flight data obtained from the flight described in the main text of the report. The data presented are a series of time histories of internal variables from TAFCOS with the addition of a number of quantities measured directly from aircraft sensors. The variables shown are grouped in accordance with the block diagram structure described in appendix A and shown in figure 4. To permit direct comparison with the TAFCOS program, each variable is labeled with the assembly language programming name. As has been previously described, the aircraft was required to fly about 2-1/2 circuits around a racetrack-type pattern. For interpretation of the data, the location of the aircraft on the pattern can be seen from either the time scale on the bottom of the plots or from the variable NWP on the top of each page. The symbol NWP stands for way point and is the next way point ahead of the aircraft where the numbers are those shown on figure 7. The data start with the aircraft approaching way point 9 on the downwind leg and end with a final approach to the runway at way point 15. For those situations in which the aircraft repeats the pattern, after passing way point 15 the NWP counter resets to NWP = 1.

Most of the data that follow are presented with a minimum of discussion. Each variable is defined and a brief description of how each fits into the TAFCOS structure is presented. The most important variables have already been discussed in the main text and many of the others explained in the appendix on the theoretical structure. The main purpose of this appendix, therefore, is to simply provide a thorough documentation of the flight.

ATCGEN Variables

The first set of plots, figure 27, gives information about the generation of the command signal from the ATC trajectory command generator. The principal quantities shown are the path length variables used to generate the position, velocity, and acceleration command inputs to the main structure of TAFCOS.

NWP — next way point

DELFLP — aircraft flap setting (The flap setting ranged from full up on the downwind leg to full down at about 37.5° on the 6° final approach. TAFCOS provides a flap command output but the aircraft does not have servo-driven flaps so that the flap changes were manually controlled by the pilot in accordance with standard flight procedures.)

SATC — commanded position of the aircraft in terms of path length along the prescribed track and measured from the way point behind the aircraft

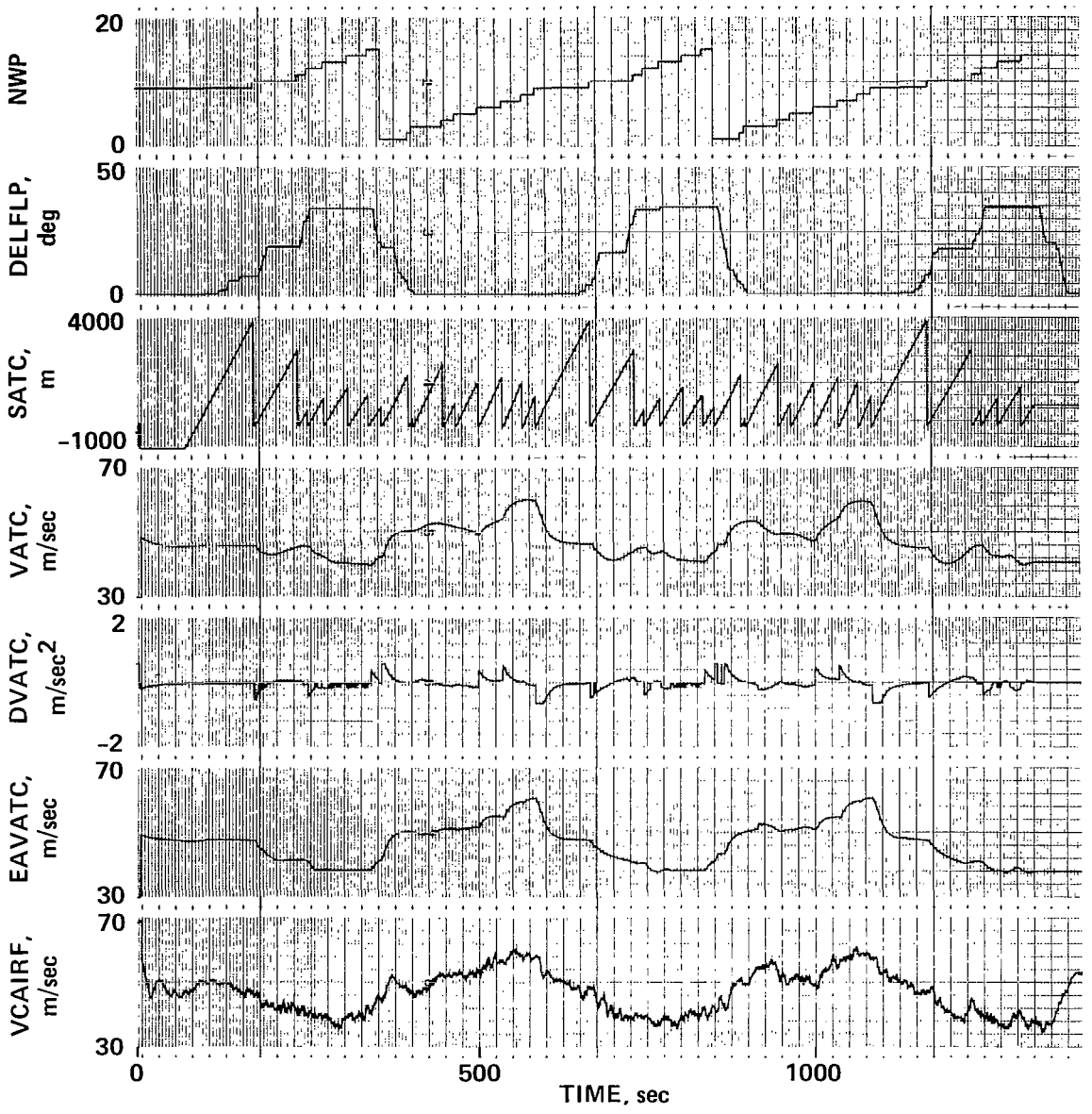


Figure 27.- ATCGEN variables.

VATC commanded velocity prescribed for the commanded aircraft position

DVATC commanded acceleration as above

EAVATC commanded air speed — VATC corrected for estimated wind from the navigation routine

VCAIRF calibrated air speed from aircraft air data sensor for comparison with the commanded airspeed

TRACOM Variables

Trajectory command generator variables, figure 28.

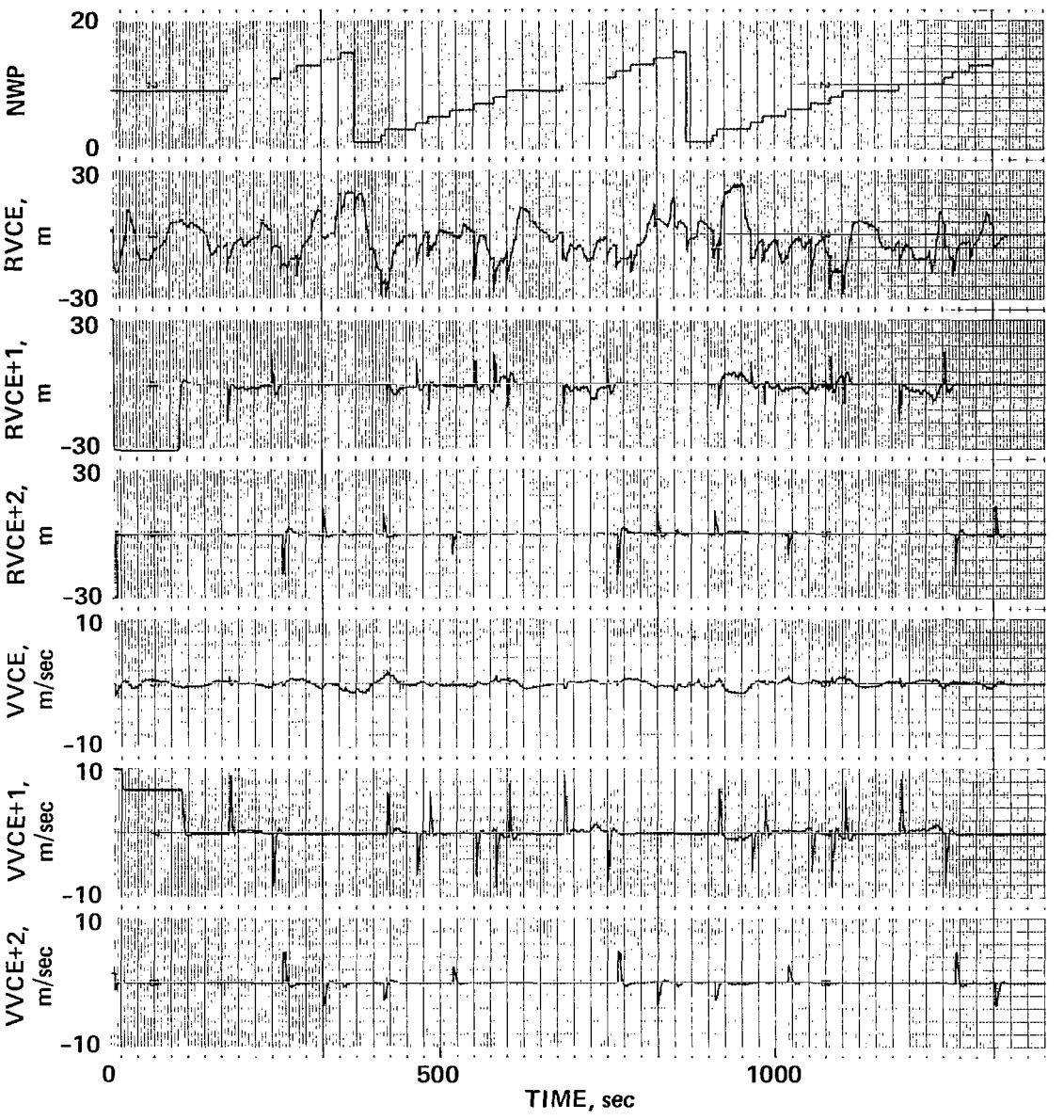
NWP — next way point

RVCE, RVCE+1, RVCE+2 — position error between the position command from ATC, the variable RSS, and the smoothed position output from TRACOM, the variable RSC (RVCE is resolved into a velocity axis system where RVCE is along the velocity vector, the +1 component the horizontal portion, and the +2 the vertical. The +1 and +2 components show spiking on the data traces due to the segment switching logic in ATCGEN. TRACOM is required to provide smoothing for these jumps. The noise on the component along the velocity vector is from the wind estimate noise due to turbulence and the consequent effect on the coordinate transformation operation.)

VVCE, VVCE+1, VVCE+2 — the velocity error associated with the position error described above

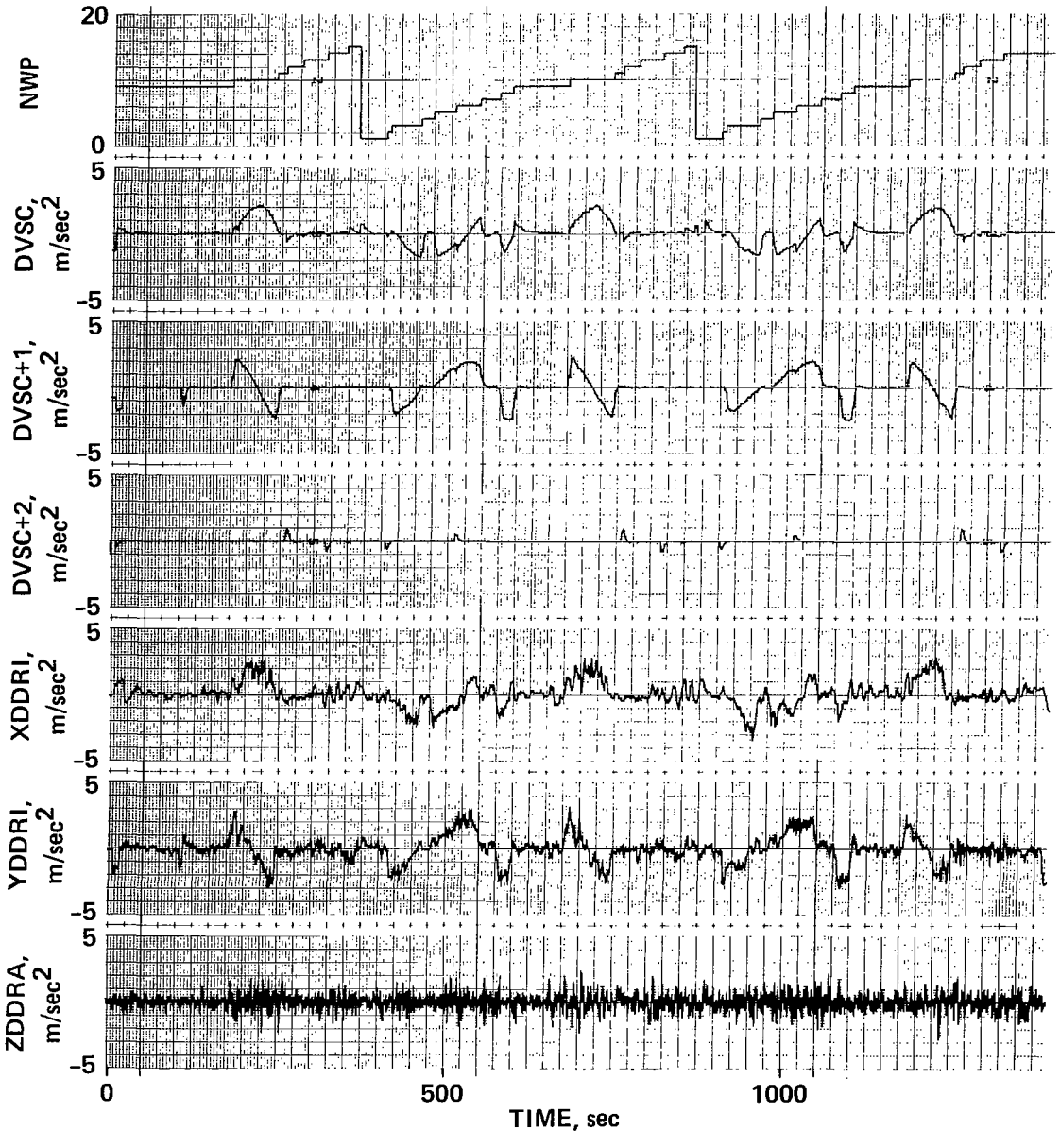
DVSC, DVSC+1, DVSC+2 — commanded acceleration output from TRACOM (The three components are defined in the ground or inertial system. This signal is the main drive signal for TAFCOS and is the acceleration required to follow the commanded path.)

XDDRI, YDDRI, ZDDRA — measured vehicle acceleration from an on-board set of accelerometers (The components are resolved in the ground coordinate system for comparison with the commanded accelerations given above.)



(a)

Figure 28.- TRACOM variables.



(b)

Figure 28.- Concluded.

TRAPCO Variables

Trajectory perturbation controller variables, figure 29.

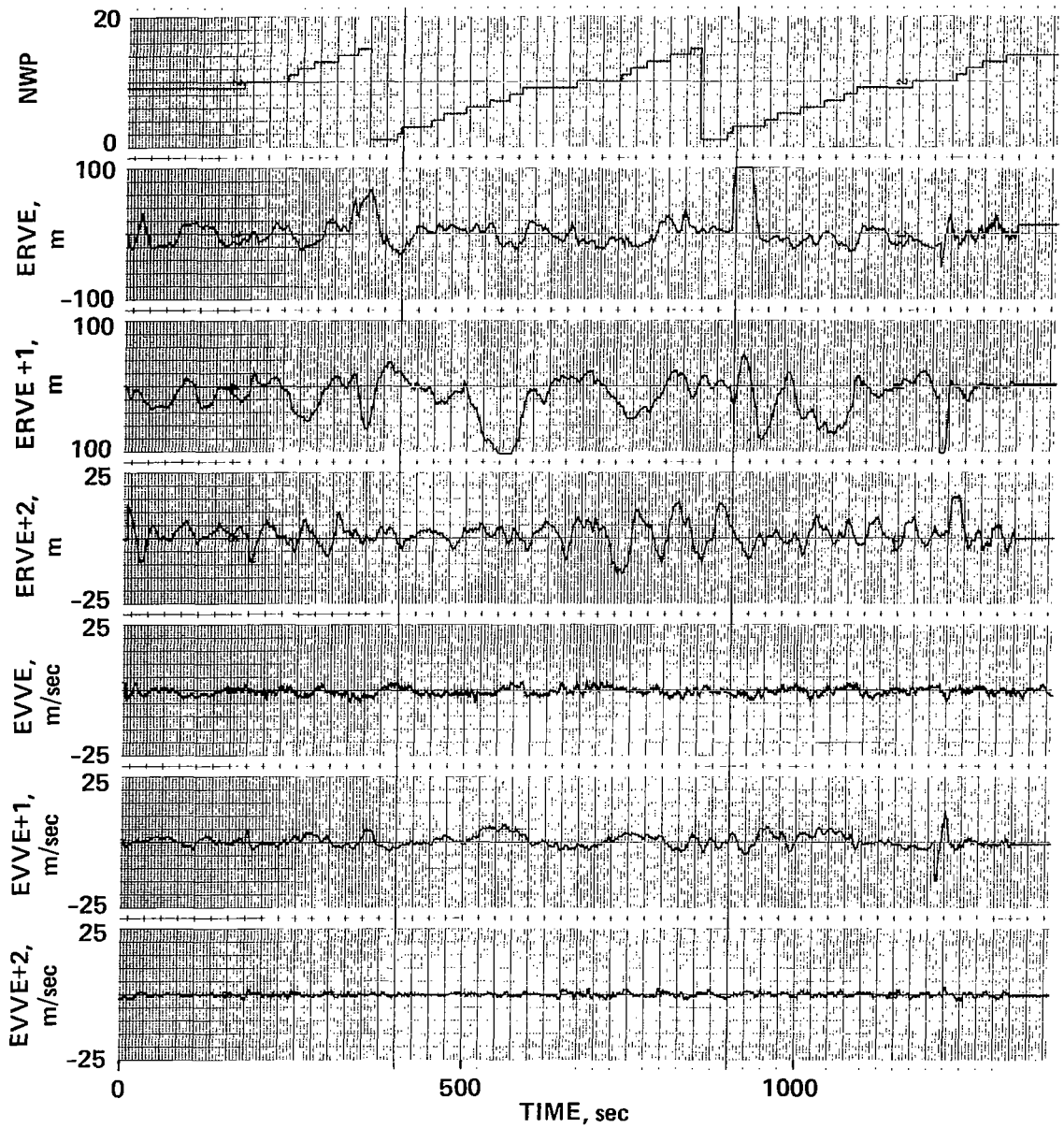
NWP — next way point

ERVE, ERVE+1, ERVE+2 — position error for feedback control (ERVE is the difference between the commanded position, RSC from TRACOM, and the estimated position is determined in the navigator, ERS. The components are in the velocity coordinate system as previously described. The second two variables are the quantities presented to the pilot on the EADI as lateral and vertical deviation.)

EVVE, EVVE+1, EVVE+2 — velocity error for feedback control defined as above

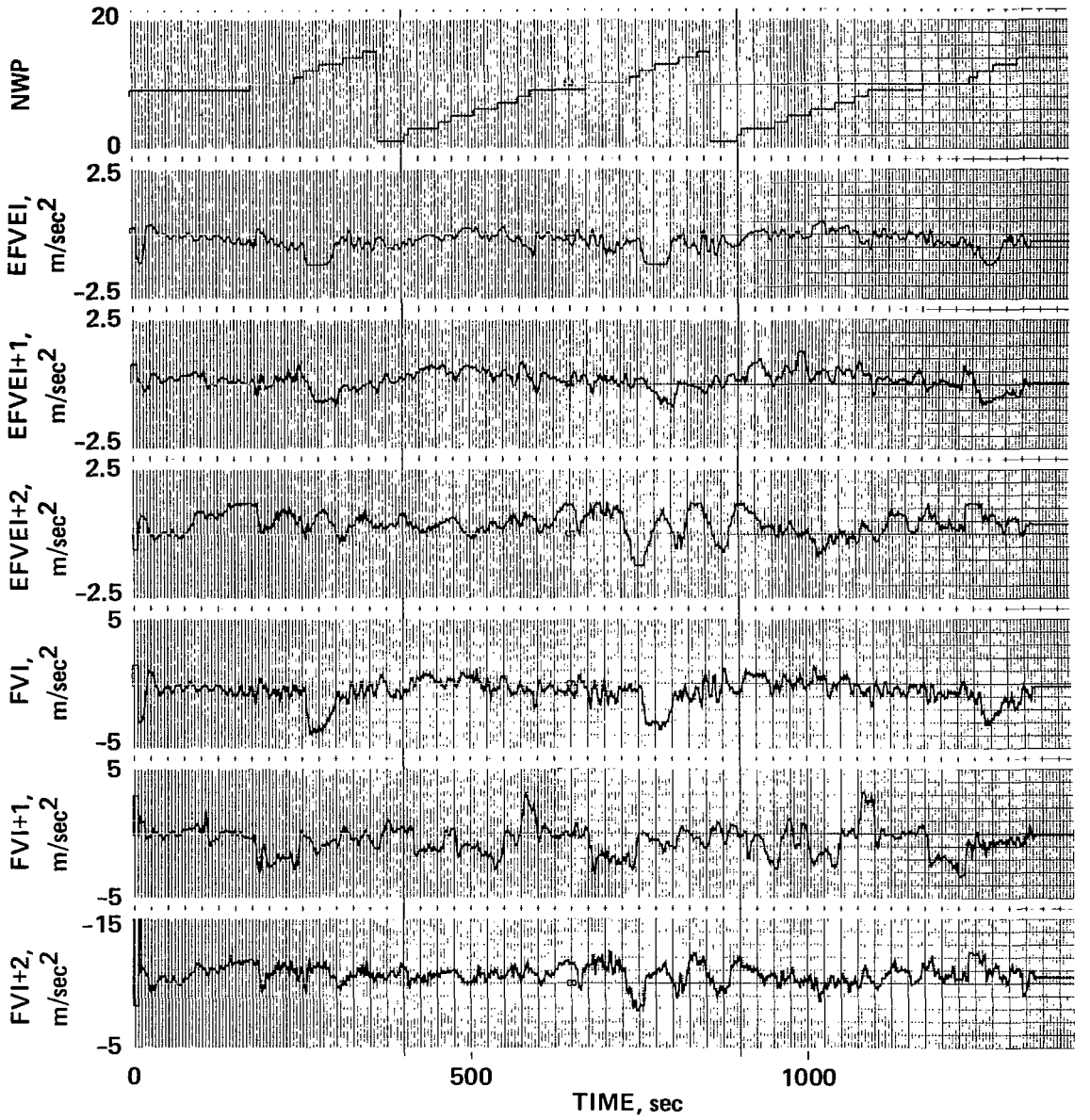
EFVEI, EFVEI+1, EFVEI+2 — integral of acceleration error (The term EFVEI is a feedback term as are the two variables given above where the value is the integral of the acceleration error between the commanded acceleration and the measured equivalent from aircraft sensors. Long-term biases primarily come from modeling errors in the trimmaps and the short-term biases are caused by wind turbulence, navigation inconsistencies, configuration changes, etc.)

FVI, FVI+1, FVI+2 — FVI is the acceleration input to the trimmap; it is proportional to the aerodynamic and thrust forces required to drive the aircraft along the commanded path (FVI is the sum of the acceleration command DVSC resolved into the velocity axis system and the three feedback commands given above.)



(a)

Figure 29.- TRAPCO variables.



(b)

Figure 29.- Concluded.

FTRIM Variables

Force trimmap variables, figure 30.

NWP — next waypoint

ANGS, ANGS+1 — ANGS and ANGS+1 are the attitude command outputs from the trimmap and represent commanded roll angle and angle of attack, respectively (These signals become the input commands to the attitude control loop, with the beta command assumed zero; they are comparable to the position commands from ATCGEN.)

THROTS — throttle handle position required to produce the thrust component of the force commanded by FVI

THROTC — commanded throttle handle position after processing through a throttle command generator to ensure an executable command by the aircraft engines (Essentially the throttle motion as seen by the pilot through the throttle handle servo system.)

THRORC — actual throttle output signal to STOLAND (THRORC is a delta command and is the difference between THROTC and the measured throttle handle position from the aircraft.)

LRPM — engine rpm from the left-hand engine (The engine rpm is manually controlled by the pilot and is usually either 75% for cruise or 100% for final approach and climb-out. The engine rpm value is used in the throttle trimmap calculations that generate the throttle commands.)

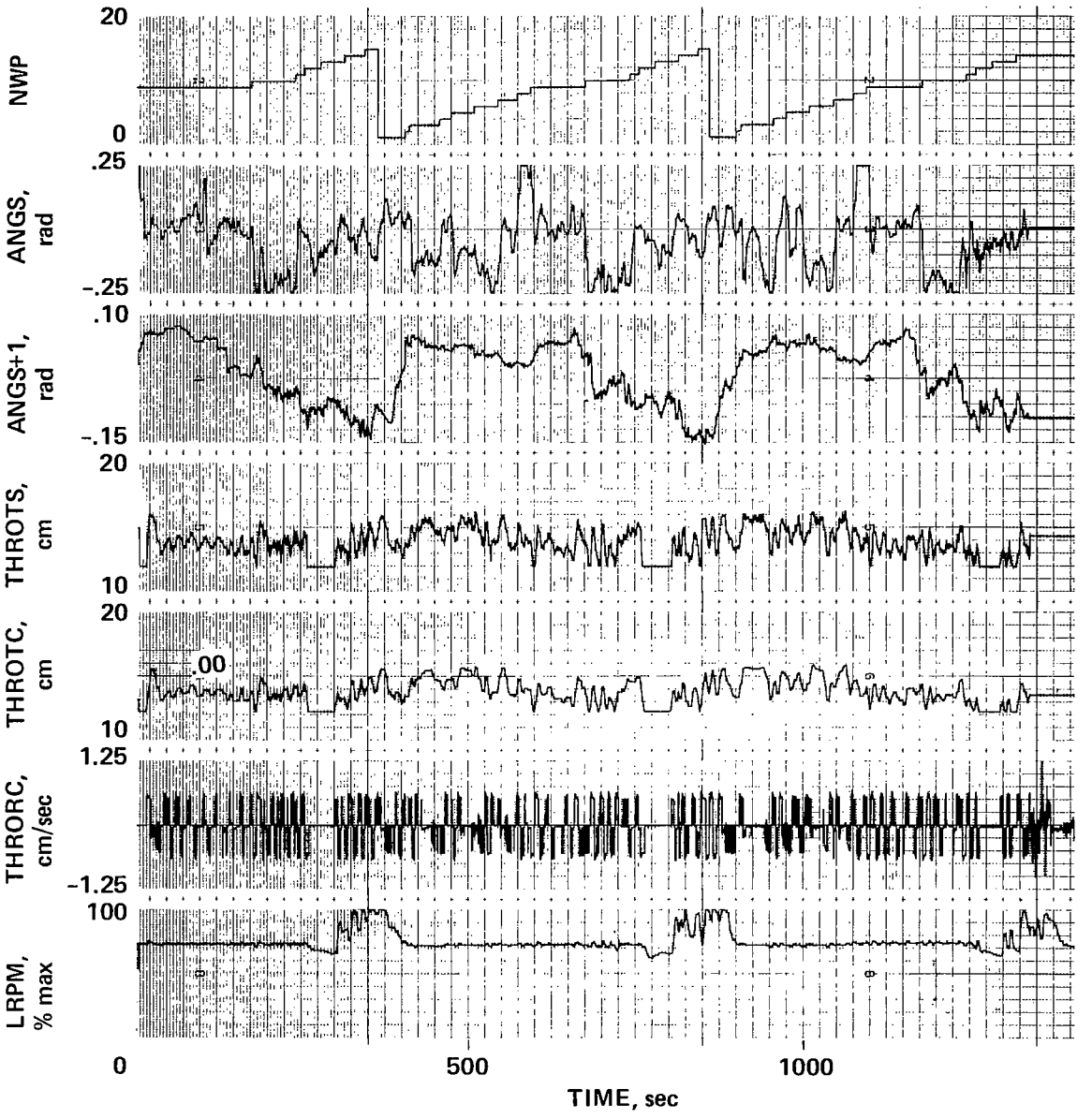
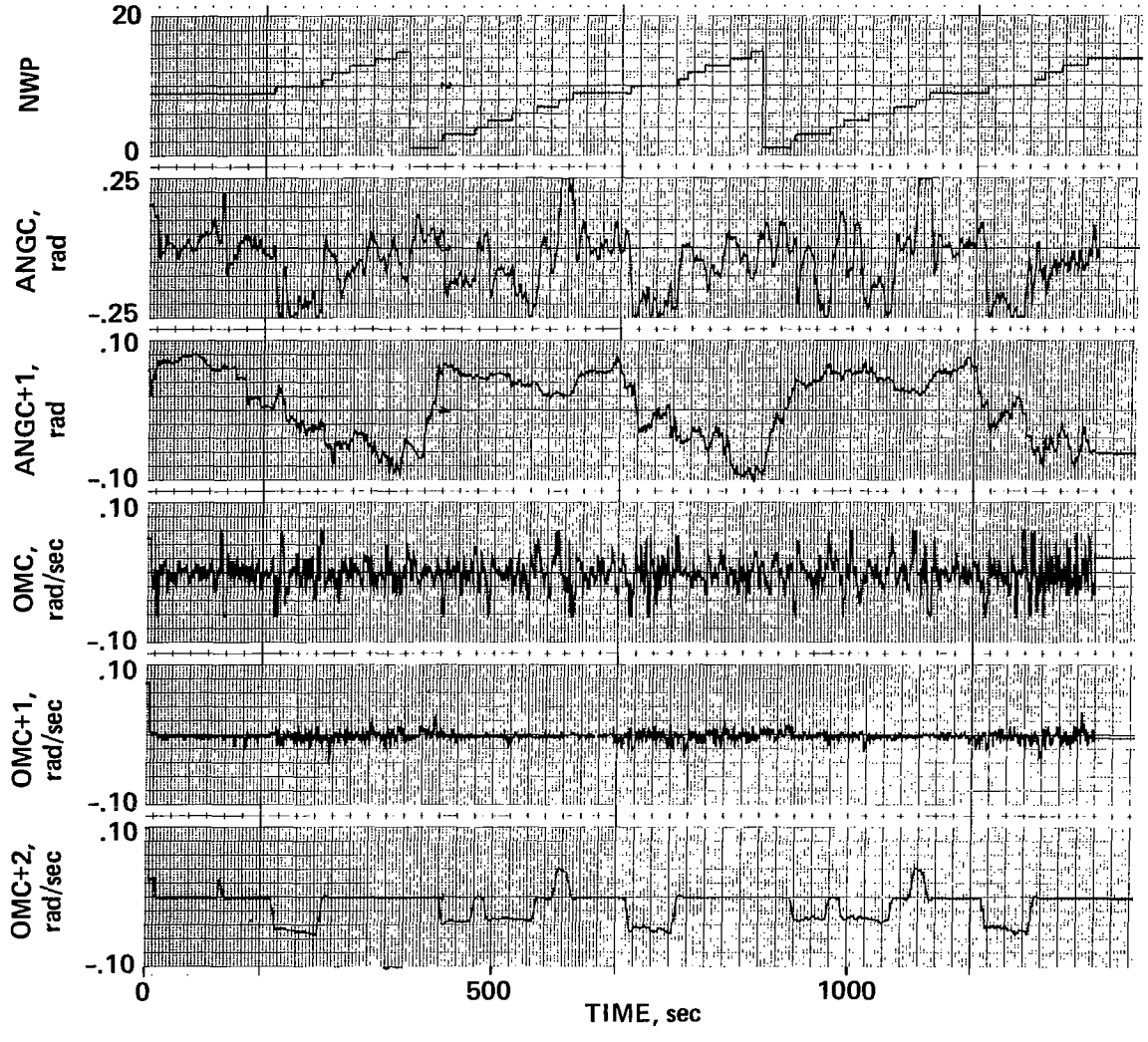


Figure 30.- FTRIM variables.

ROTCOM Variables

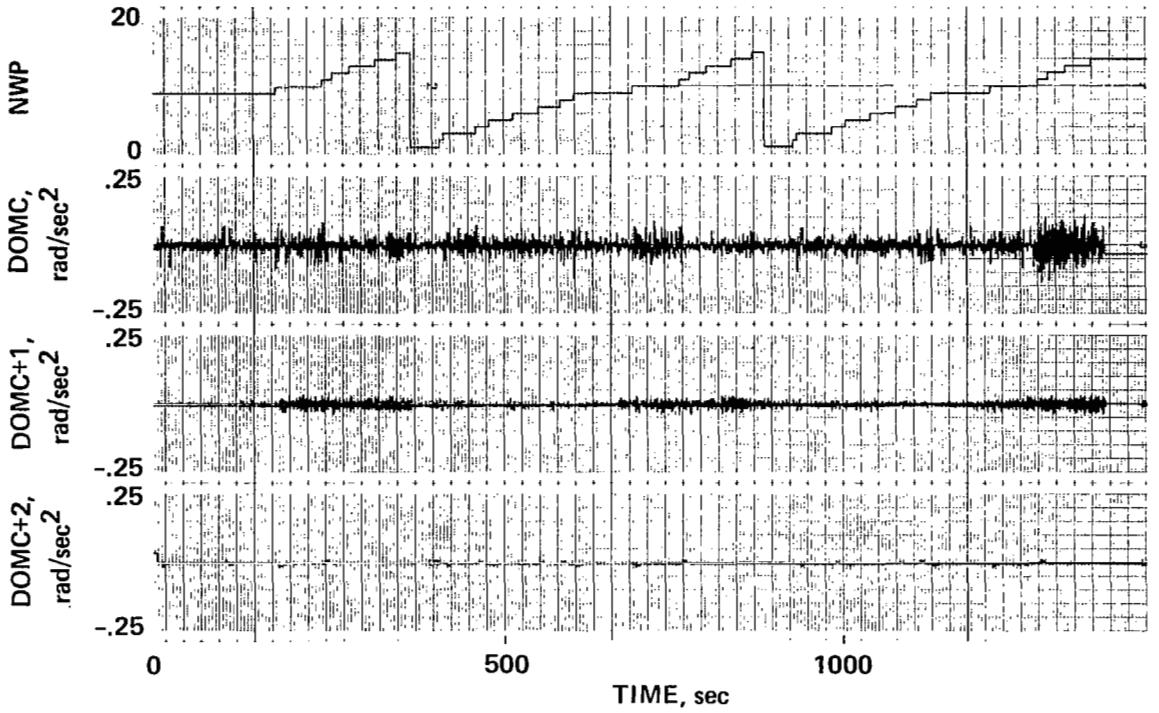
Rotation command generator variables, figure 31.

NWP	next way point
ANGC ANGC+1	because ROTCOM performs a function similar to that of the trajectory command generator, ANGC and ANGC+1 are the flyable version of ANGS and ANGS+1 and represent the commanded roll angle and angle of attack (ANGC and ANGC+1 are nearly identical to ANGS and ANGS+1, indicating that the trajectory command loop is asking for attitude response that is flyable.)



(a)

Figure 31.- ROTCOM variables.



(b)

Figure 31.- Concluded.

OMC, OMC+1, OMC+2 — the OMC variables are the commanded angular rate in the aircraft body coordinate system (These values are internally generated within ROTCOM and along with the angular acceleration form the total input command to the attitude loop.)

DOMC, DOMC+1, DOMC+2 — computed angular acceleration command

ROTPCO Variables

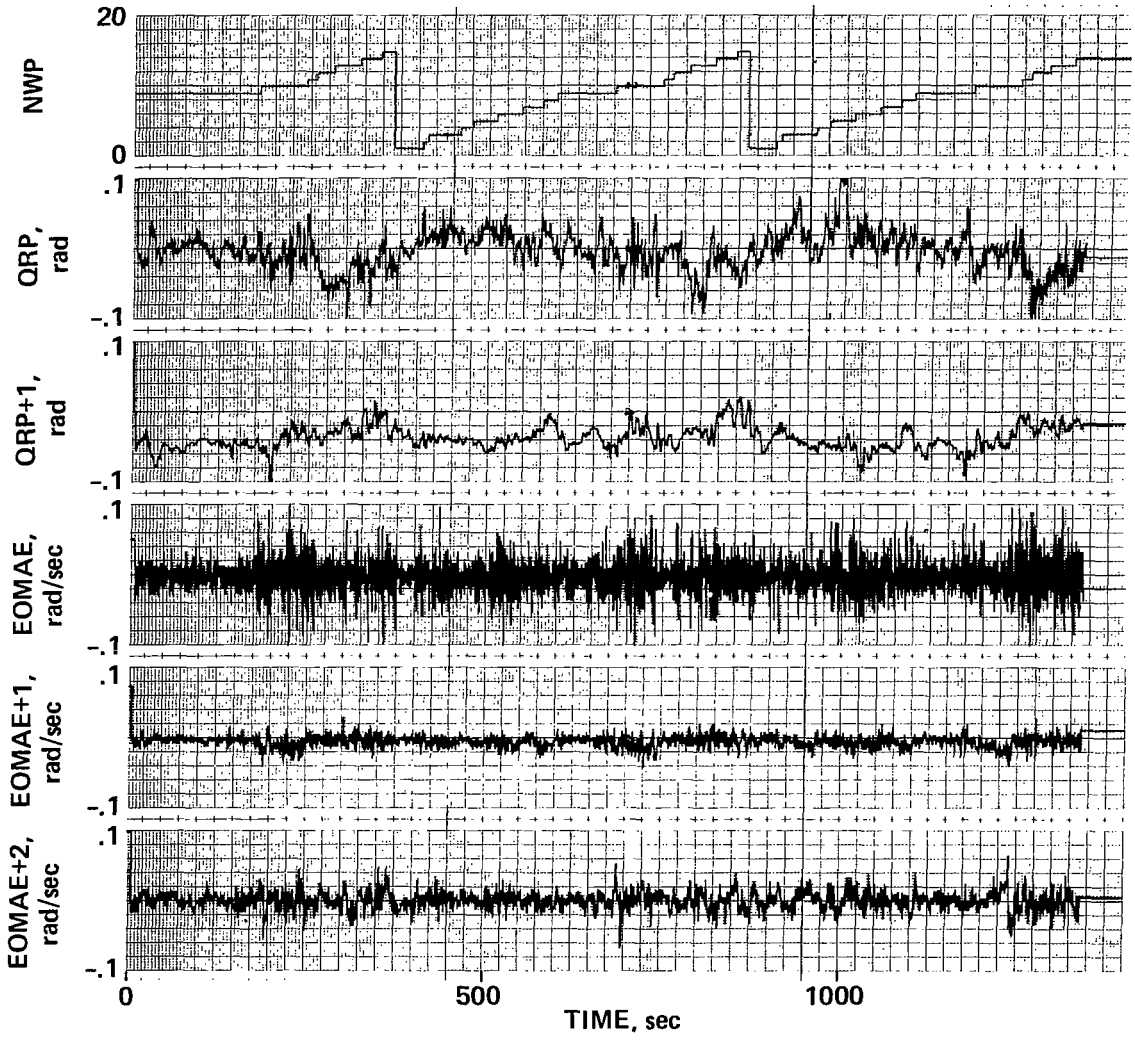
Rotation perturbation controller variables, figure 23.

NWP — next way point

QRP, QRP+1 — attitude feedback error (The quantities QRP and QRP+1 are the difference between the variables ANGC and ANGC+1 and the equivalent quantities derived from aircraft attitude and rate measurements.)

EOMAE
EOMAE+1
EOMAE+2 } angular rate feedback error

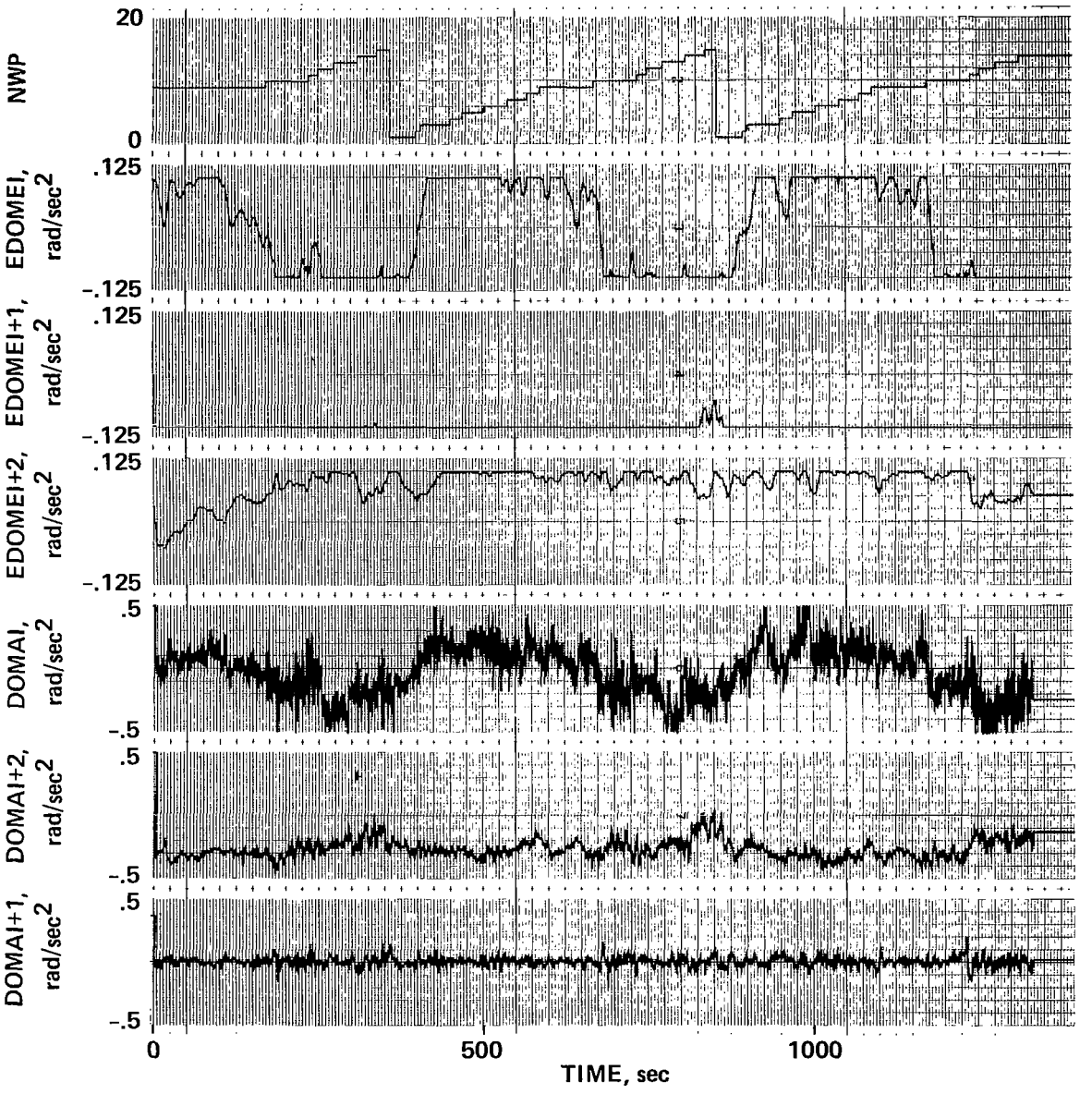
EDOMEI
EDOMEI+1
EDOMEI+2 } integral of angular rate error (The quantities are the integral of the commanded angular rate and a derived angular rate. No measurement of angular rate is available directly from the aircraft sensors. Note the saturation of all three quantities and the biases on the second and third. The integral limits were set intentionally low so that the saturation effects appear stronger than they really are. The biasing, however, does indicate modeling errors in the moment trimmaps, especially the pitch axis, with a need for improved aircraft parameters.)



(a)

Figure 32.- ROTPCO variables.

DOMAI
DOMAI+1
DOMAI+2
} angular acceleration command for trimmap input (DOMAI is the sum of the acceleration command from ROTCOM plus the sum of the three feedbacks given above.)



(b)

Figure 32.- Concluded.

Control Surface Positions

The three quantities presented in figure 33 are the actual measured control surface positions in response to the commands from the moment trimmap. Examination of other data shows that the servo errors are small and that these quantities are essentially identical to the commanded control surface values.

AILPOS aileron position, positive for positive roll moment

ELVPOS elevator position, positive for positive pitching moment

RUDPOS rudder position, positive for positive yaw moment

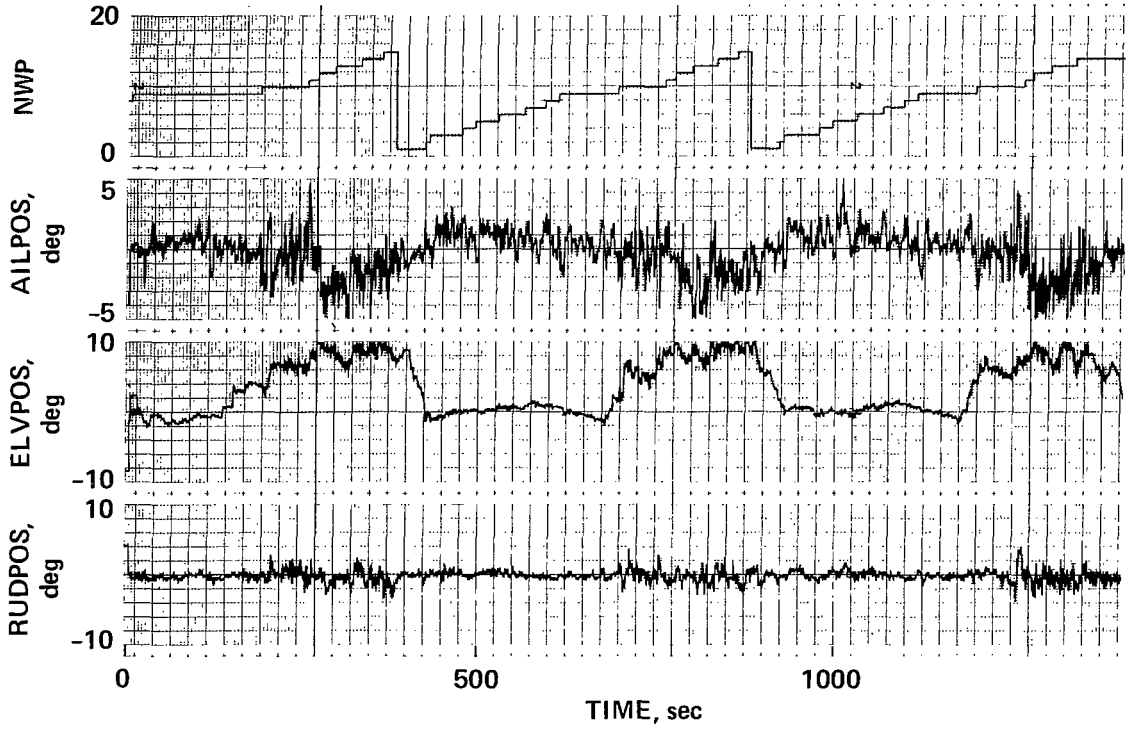


Figure 33.- Control surface positions.

Aircraft Attitude and Rate Variables

Figure 34 presents a set of measured attitude and rate quantities for the aircraft. They are not directly associated with the TAFCOS controller but simply show a time history of aircraft attitude response to the commands.

PHI roll angle

THETA pitch angle

PSIA heading angle

PHIDOT roll angular rate — body axis

THTDOT pitch angular rate — body axis

PSIDOT yaw angular rate — body axis

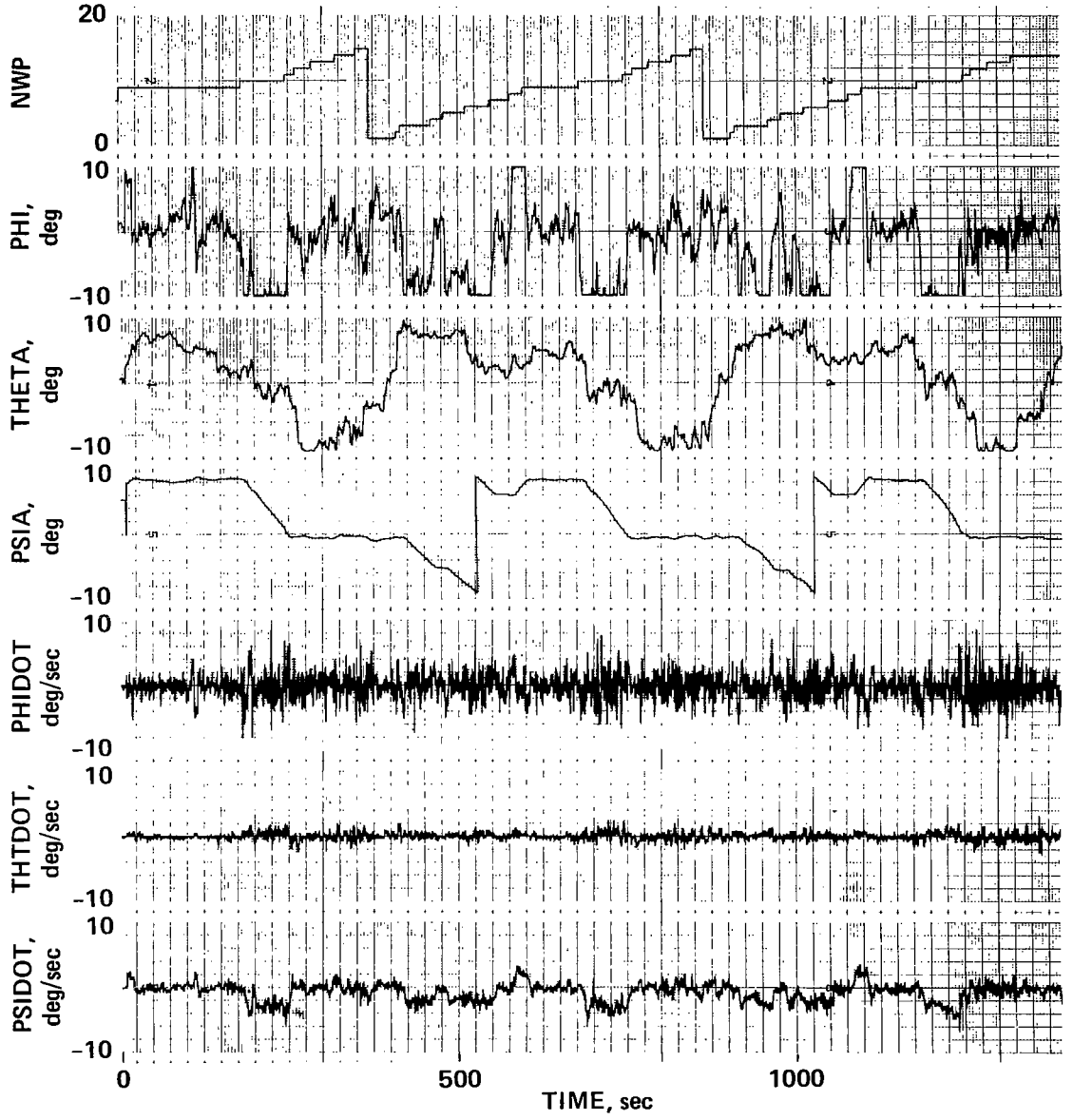


Figure 34.- Aircraft attitude and rate variables.

REFERENCES

1. Meyer, George; and Cicolani, Luigi S.: A Formal Structure for Advanced Automatic Flight Control Systems. NASA TN D-7940, 1975.

2. Grgurich, John; and Bradbury, Peter: STOLAND Final Report, NASA CR-137972, 1976.

3. Neuman, Frank; Watson, Delamar M.; and Bradbury, Peter: Operational Description of an Experimental Digital Avionics System for STOL Airplanes. NASA TM X-62,448, 1975.

4. Wehrend, William R., Jr.: Pilot Control Through the TAFCOS Automatic Flight Control System. NASA TM-81152, 1979.

5. Smith, G. Allen; and Meyer, George: Application of the Concept of Dynamic Trim Control to Automatic Landing of Carrier Aircraft. NASA TP-1512, 1979.

6. Wehrend, William R., Jr.; and Meyer, George: Program Listing for the TAFCOS DHC-6 Flight Test. NASA TM-78607, 1979.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-1513		

4. Title and Subtitle	5. Report Date
FLIGHT TESTS OF THE TOTAL AUTOMATIC FLIGHT CONTROL SYSTEM (TAFCOS) CONCEPT ON A DHC-6 TWIN OTTER AIRCRAFT	February 1980
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
William R. Wehrend, Jr. and George Meyer	A-7901
	10. Work Unit No. 505-07-31

9. Performing Organization Name and Address	11. Contract or Grant No.
Ames Research Center, NASA Moffett Field, Calif. 94035	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Paper
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The newer aircraft designs — especially the STOL and VTOL designs, and those using active control concepts — call for a new generation of flight control systems capable of handling the more complex operational requirements. Over the past few years, a new flight control concept called TAFCOS (total automatic flight control system) has been under development at Ames Research Center. Performance of the system concept, which provides a workable structure for satisfactory control of these advanced aircraft, has recently been verified in flight test. This paper presents an overview of the controller structure, a description of the mechanization for flight test, and a presentation of flight data. The results of the flight test show that the integrated structure of TAFCOS provides for a programming structure that is compact, and one in which space and time requirements are well below those of conventional designs. The flight performance also demonstrated that TAFCOS can provide good control of the aircraft over a wide spectrum of the flight envelopes.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Handling qualities Flight controls Autopilot	Unlimited STAR Category - 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	73	$5.25

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1980

2 1 1U,A, 020280 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LABORATORY
ATTN: TECHNICAL LIBRARY (SUL)
KIRTLAND AFB NM 87117